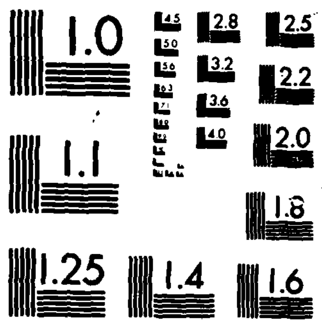
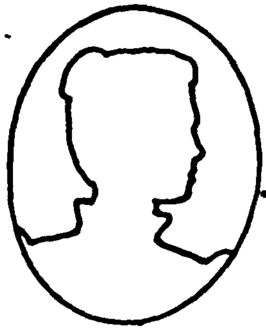
AD-A165 350   ADA (TRADEMARK) TRAINING CURRICULUM INSTRUCTOR'S COURSE   1/2
S500 OVERVIEW(U) SOFTECH INC WALTHAM MA 1986
DAAB07-85-C-K506

UNCLASSIFIED   F/G 5/9   NL


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

AD-A165 350 0

1986



Ada® Training Curriculum

AD-A165 350

86 3 11 151

Instructor's Course
S500
Overview

U.S. Army Communications-Electronics Command
(CECOM)

Contract DAAB07-85-C-K506

Prepared By:

SOFTECH, INC.
460 Totten Pond Road
Waltham, MA 02154

•Ada is a registered trademark of the U.S. Government, Ada Joint Program Office

SOFTECH

INSTRUCTOR'S COURSE MODULE (S500)

931/A - OVERVIEW

VG 931/A

46© Totten Pond Road, Waltham, MA 02254 (617)890-6900 TWX:710-324-6401

-1-

INSTRUCTOR'S COURSE OUTLINE

VG 931/A

This INSTRUCTOR'S COURSE consists of the following sections:

OVERVIEW

(1) PHILOSOPHY BEHIND THE CURRICULUM;

(2) OVERVIEW OF THE CURRICULUM; and

(3) GENERAL CONSIDERATIONS. Keywords: ADA programming language, training, instruction manuals.

VG 931/A

INSTRUCTOR NOTES

- ALLOW 45 MINUTES FOR THIS SECTION
- SEE INSTRUCTOR'S NOTE 1-2i FOR SUGGESTIONS ABOUT CLASS INTRODUCTIONS

SECTION 1

PHILOSOPHY BEHIND THE CURRICULUM

INSTRUCTOR NOTES

- MAKE SURE INSTRUCTORS IN TRAINING UNDERSTAND THAT THIS COURSE WILL NOT TEACH Ada, METHODOLOGY OR ENVIRONMENT.
- ALSO MAKE SURE INSTRUCTORS IN TRAINING UNDERSTAND THIS COURSE ASSUMES THEY ARE EXPERIENCED TEACHERS.

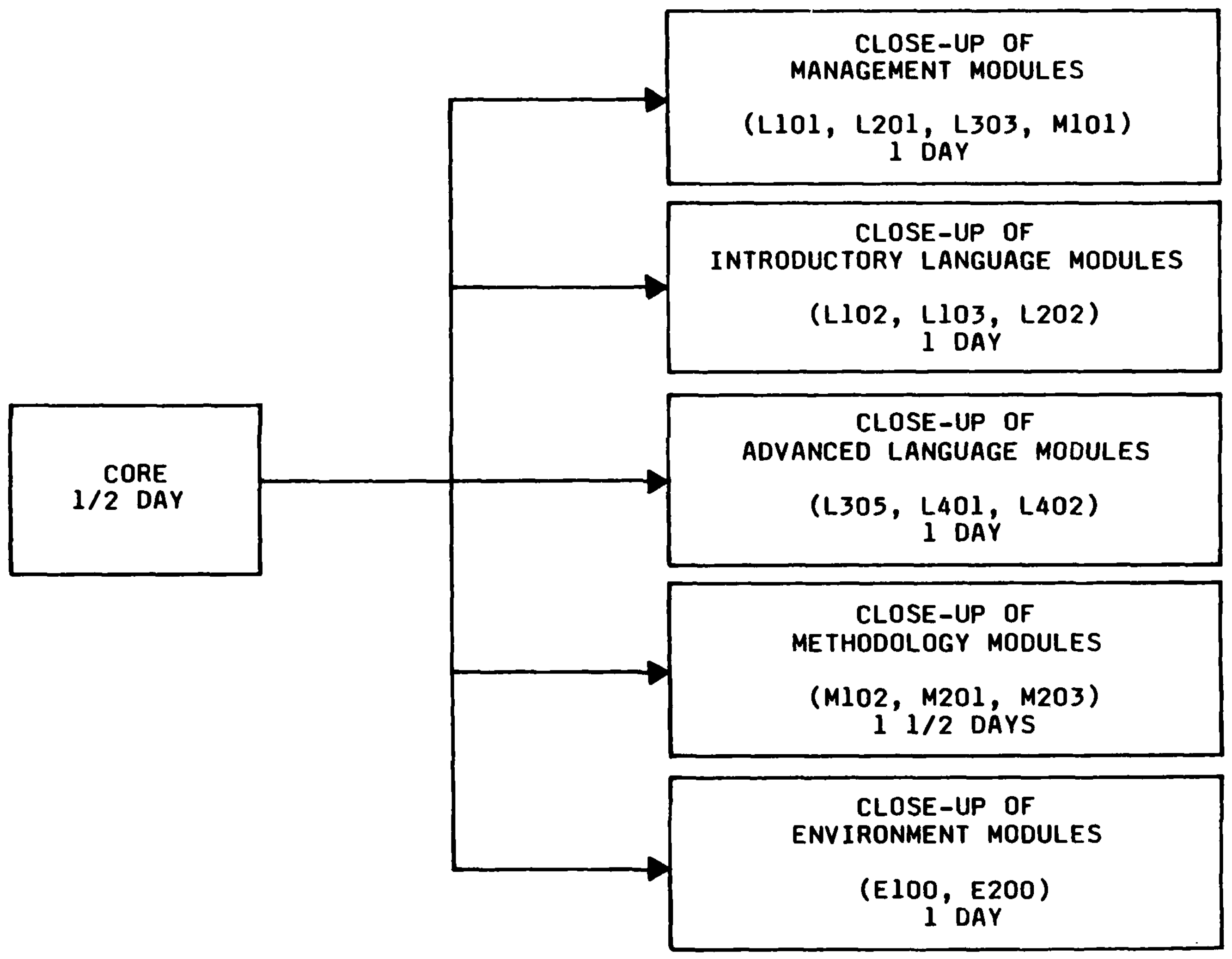
ABOUT THIS COURSE

- COURSE FOR INSTRUCTORS
  - EXPERIENCED TEACHERS
  - UNDERSTANDING OF MATERIAL IN MODULES TO BE TAUGHT
- COURSE DOES NOT TEACH ABOUT
  - Ada
  - METHODOLOGY
  - ENVIRONMENT
  - PEDAGOGIC METHODS
- UPON COMPLETION YOU WILL UNDERSTAND
  - BASIC STRUCTURE OF Ada CURRICULUM AND PHILOSOPHY BEHIND IT
  - ROLE OF EACH MODULE IN THE CURRICULUM
  - BACKGROUND EXPECTED OF STUDENTS STUDYING A PARTICULAR MODULE
  - WHERE IN THE CURRICULUM VARIOUS TOPICS ARE COVERED
  - GOALS OF EACH MODULE IN THE CURRICULUM
  - OVERALL STRUCTURE OF EACH MODULE IN CURRICULUM
  - ROLE OF EACH MODULE'S EXERCISES
  - WHERE APPLICABLE
    - SPECIAL CONSIDERATIONS WHEN TEACHING CERTAIN MODULES, INCLUDING
      - COMMON SOURCES OF CONFUSION
      - COMMON AREAS OF STUDENT RESISTANCE
    - APPROACHES THAT HAVE PROVEN TO BE EFFECTIVE OR INEFFECTIVE IN CONVEYING CERTAIN POINTS

INSTRUCTOR NOTES

- MAKE SURE CLASS UNDERSTANDS HOW THIS COURSE IS STRUCTURED.
- THIS IS A GOOD PLACE TO GET THE INSTRUCTORS IN TRAINING TO INTRODUCE THEMSELVES AND DESCRIBE
  - WHAT THEY EXPECT TO BE TEACHING
  - Ada BACKGROUND AND/OR METHODOLOGY BACKGROUND
  - TEACHING BACKGROUND

COURSE STRUCTURE


CORE
1/2 DAY
CLOSE-UP OF MANAGEMENT MODULES
(L101, L201, L303, M101)
1 DAY
CLOSE-UP OF INTRODUCTORY LANGUAGE MODULES
(L102, L103, L202)
1 DAY
CLOSE-UP OF ADVANCED LANGUAGE MODULES
(L305, L401, L402)
1 DAY
CLOSE-UP OF METHODOLOGY MODULES
(M102, M201, M203)
1 1/2 DAYS
CLOSE-UP OF ENVIRONMENT MODULES
(E100, E200)
1 DAY

INSTRUCTOR NOTES

- THE JOB CLASSIFICATIONS SHOWN ON THIS SLIDE WERE DETERMINED IN A STUDY PERFORMED BY SOFTECH FOR CECOM (Ada SOFTWARE DESIGN METHODS FORMULATION OCTOBER 1982, CONTRACT NO. DAAK80-80-C-0187)
- STUDY INCLUDED
  - INDUSTRY/GOVERNMENT WORK FORCE STUDY
    - 10 COMPANIES/GOVERNMENT AGENCIES
    - 428 RESPONSES (OUT OF 720)
  - QUESTIONS ASKED ABOUT
    - TECHNICAL BACKGROUND
    - PRINCIPAL OUTPUTS AND DUTIES
    - KNOWLEDGE OF PROGRAMMING LANGUAGES, SOFTWARE METHODOLOGIES, PROGRAMMING CONCEPTS
- EMPHASIZE THAT ORGANIZATION OF CURRICULUM AND TARGET AUDIENCE WAS WELL-PLANNED, NOT POT LUCK.
- THE JOB CLASSIFICATIONS IN THE OTHERS CATEGORY CONSTITUTES A LARGE MIX. PEOPLE IN THIS CATEGORY MAY
  - CONDUCT DESIGN REVIEW, CODE WALKTHROUGHS, REQUIREMENT REVIEW
  - PERFORM SYSTEM ANALYSIS, PROGRAM MANAGEMENT, QUALITY ASSURANCE,

    CONFIGURATION MANAGEMENT
  - FORMULATE POLICY AND STRATEGY
  - PROVIDE MARKETING SUPPORT

  SO THEIR NEEDS FOR TRAINING VARY GREATLY.

WHO NEEDS TO BE TRAINED AND WHAT DO THEY NEED TO KNOW?

- PROJECT/TASK LEADERS
  - CONCEPTUAL UNDERSTANDING OF FULL Ada
  - SOFTWARE ENGINEERING
  - PROGRAMMING METHODOLOGY
- DESIGN CONSULTANTS
  - FULL Ada
  - SOFTWARE ENGINEERING METHODOLOGIES
  - PROGRAMMING METHODOLOGY
- REAL-TIME SYSTEM ARCHITECTS
  - FULL Ada
  - SOFTWARE ENGINEERING
  - PROGRAMMING METHODOLOGY
  - ENVIRONMENTS
- SOFTWARE DESIGNERS
  - ADVANCED Ada
  - SOFTWARE ENGINEERING
  - PROGRAMMING METHODOLOGY
  - ENVIRONMENT
- PROGRAMMERS
  - BASIC Ada
  - SOFTWARE ENGINEERING
  - PROGRAMMING METHODOLOGY
  - ENVIRONMENT
- OTHERS
  - INCLUDES: CONFIGURATION MANAGEMENT/QA ENGINEERS, SYSTEM INTEGRATION STAFF, MANAGERS, ETC.
  - NEEDS: DEPENDS ON TECHNICAL DUTIES

INSTRUCTOR NOTES

- EMPHASIZE THAT THE MODULES ARE COURSE BUILDING BLOCKS.
- INSTRUCTORS CAN BUILD COURSES TO SATISFY THE NEEDS OF THE INTENDED AUDIENCE.
- EMPHASIZE THE NEED TO FOLLOW PREREQUISITES.

COURSES CONSIST OF MODULES

- CURRICULUM CONSISTS OF MODULES NOT COURSES

- COURSES ARE BUILT UP FROM MODULES
  - COMBINATION OF
    - Ada LANGUAGE MODULES
    - METHODOLOGY MODULES
    - ENVIRONMENT MODULES
  - COURSE TAILORED FOR INTENDED AUDIENCE
    - JOB: Technical Manager, Programmer, Real-Time System Designer
    - HIGH LEVEL LANGUAGE EXPERIENCE: None, FORTRAN only, Pascal/Modula
    - METHODOLOGY BACKGROUND: None, PDL, SREM, Structured Design
    - ENVIRONMENT: will not use (e.g. technical manager, design consultant), programmer, real-time system architect

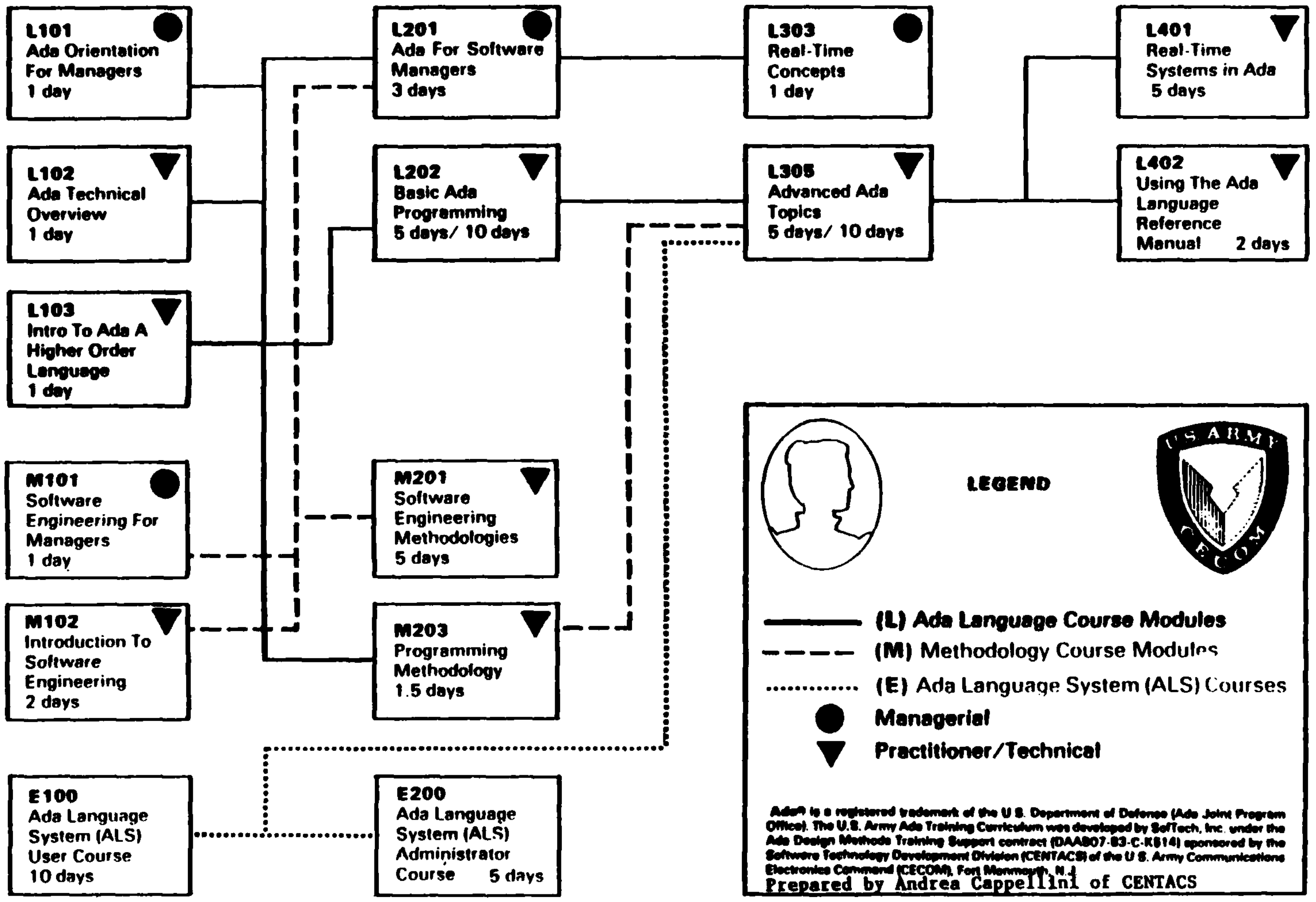
- PREREQUISITES ARE IMPORTANT
  - INDICATE EXPERTISE REQUIRED TO UNDERSTAND/APPRECIATE MODULE
  - EXAMPLE:

    PROGRAMMERS/DESIGNERS WHO HAVE NEVER PROGRAMMED IN A HOL SHOULD NOT BE THROWN INTO THE <u>Basic Ada Programming</u> MODULE WITHOUT FIRST GOING THROUGH THE <u>Introduction to Ada-A Higher Order Language</u> MODULE.

INSTRUCTOR NOTES

- TRACE THROUGH A TYPICAL COURSE
  - M102 AND L102 (CAN BE TAUGHT IN PARALLEL)
  - M203 AND L202 (CAN BE TAUGHT IN PARALLEL)
  - L305
  - L401
- POINT OUT THAT EACH COLUMN CORRESPONDS TO A DISTINCT LEVEL.
- EXPLAIN THE AND/OR GRAPHS.
- LET THE CLASS KNOW THAT WE WILL GIVE AN OVERVIEW OF EACH MODULE LATER IN THE CORE PART OF THE COURSE.

U.S. ARMY Ada TRAINING CURRICULUM


L101 Ada Orientation For Managers 1 day
L102 Ada Technical Overview 1 day
L103 Intro To Ada A Higher Order Language 1 day
M101 Software Engineering For Managers 1 day
M102 Introduction To Software Engineering 2 days
E100 Ada Language System (ALS) User Course 10 days
L201 Ada For Software Managers 3 days
L202 Basic Ada Programming 5 days/ 10 days
M201 Software Engineering Methodologies 5 days
M203 Programming Methodology 1.5 days
E200 Ada Language System (ALS) Administrator Course 5 days
L303 Real-Time Concepts 1 day
L305 Advanced Ada Topics 5 days/ 10 days
L401 Real-Time Systems in Ada 5 days
L402 Using The Ada Language Reference Manual 2 days
LEGEND
US ARMY
CECOM
(L) Ada Language Course Modules
(M) Methodology Course Modules
(E) Ada Language System (ALS) Courses
Managerial
Practitioner/Technical

Ada® is a registered trademark of the U.S. Department of Defense (Ada Joint Program Office). The U.S. Army Ada Training Curriculum was developed by SofTech, Inc. under the Ada Design Methods Training Support contract (DAAB07-83-C-K514) sponsored by the Software Technology Development Division (CENTACS) of the U.S. Army Communications Electronics Command (CECOM), Fort Monmouth, N.J.

Prepared by Andrea Cappellini of CENTACS

INSTRUCTOR NOTES

- THE REMAINING SLIDES DESCRIBE WHY Ada AND METHODOLOGY GO TOGETHER.

- BULLET #1 - INSTRUCTORS IN TRAINING NEED TO BE "REMINDED" WHY Ada AND SOFTWARE ENGINEERING ARE TAUGHT TOGETHER. THEY NEED TO BE REMINDED OF THE PROBLEMS WITH EXISTING SYSTEMS AND WHAT SOFTWARE ENGINEERING CONCEPTS CAN BE USED TO ADDRESS THESE PROBLEMS. THESE CONCEPTS WILL BE DIRECTLY RELATED TO Ada FEATURES IN A FEW SLIDES.

- A KEY POINT TO GET ACROSS IS THAT THE Ada LANGUAGE IS A DIRECT RESPONSE TO THE SOFTWARE CRISIS, AND AS SUCH, MANY KEY SOFTWARE ENGINEERING CONCEPTS WERE INCORPORATED INTO ITS DESIGN. THEREFORE, INSTRUCTORS IN TRAINING SHOULD MAKE THEIR CLASSES AWARE OF WHERE THESE SOFTWARE ENGINEERING CONCEPTS APPEAR IN Ada.

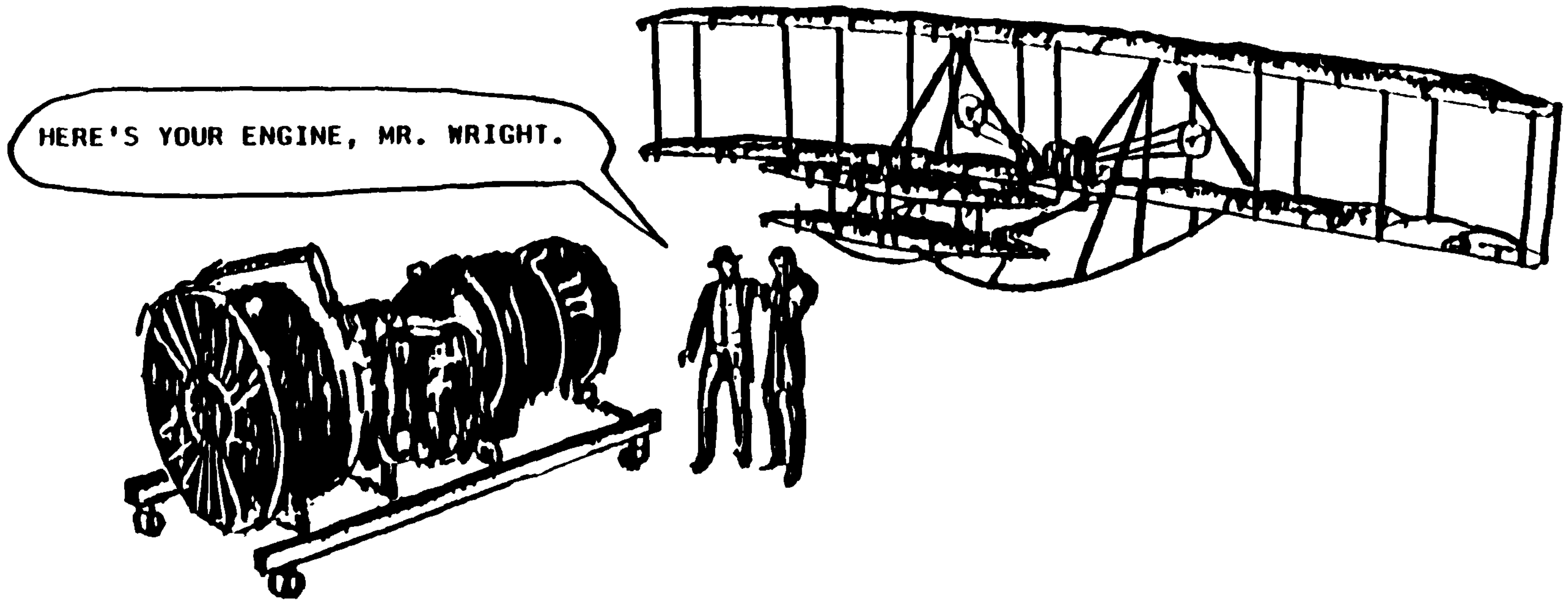
Ada AND SOFTWARE ENGINEERING

- DEVELOPMENT OF Ada IS A DIRECT RESPONSE TO SOFTWARE CRISIS
  - SYSTEMS DEVELOPED OFTEN
    - FAILED TO MEET FUNCTIONAL REQUIREMENTS
    - UNRELIABLE
    - COST TOO MUCH TO DEVELOP
    - NOT DELIVERED ON TIME
    - MODIFICATIONS COST TOO MUCH AND ERROR PRONE
    - NOT TRANSPORTABLE
    - FAILED TO MEET PERFORMANCE REQUIREMENTS
  - SOFTWARE ENGINEERING CONCEPTS ADDRESS THESE PROBLEMS
    - MODULARITY
    - INFORMATION HIDING
    - LOCALIZATION
    - COHESION
    - COUPLING
    - DATA ABSTRACTION
    - STRUCTURED PROGRAMMING
- DESIGN OF Ada INCORPORATED SOFTWARE ENGINEERING CONCEPTS
  - WHEN TEACHING Ada MODULES SHOW HOW Ada FEATURES SUPPORT SOFTWARE ENGINEERING CONCEPTS
  - WHEN TEACHING METHODOLOGY MODULES RELATE SOFTWARE ENGINEERING CONCEPTS TO Ada FEATURES SUPPORTING THEM

INSTRUCTOR NOTES

- SOME OF THE INSTRUCTORS IN TRAINING HAVE SEEN THIS SLIDE IN M201. WHILE IT IS INCLUDED FOR COMIC RELIEF, IT DOES MAKE A MAJOR POINT. Ada BY ITSELF WILL NOT SOLVE THE SOFTWARE CRISIS. IF REQUIREMENTS ARE NOT FEASIBLE AND/OR THE SOFTWARE DOES NOT SATISFY THE REQUIREMENTS, THEN Ada WILL NOT HELP.

Ada WITHOUT SOFTWARE ENGINEERING



INSTRUCTOR NOTES

- THIS SLIDE SHOWS SOME STANDARD SOFTWARE ENGINEERING CONCEPTS AND HOW THEY ARE REALIZED IN Ada. THE NEXT SLIDE TAKES THE OPPOSITE VIEW. IT SHOWS THESE SAME Ada FEATURES AND SHOWS THE SOFTWARE ENGINEERING CONCEPTS SUPPORTED BY THE FEATURES.
- INSTRUCTOR'S TEACHING METHODOLOGY MODULES SHOULD USE THIS SLIDE AS A GUIDE FOR RELATING SOFTWARE ENGINEERING CONCEPTS TO Ada FEATURES. IT IS AN IMPORTANT PART OF A METHODOLOGY INSTRUCTOR'S JOB TO MAKE THEIR STUDENTS AWARE OF HOW THESE CONCEPTS CAN BE REALIZED IN Ada.
- FOR EXAMPLE, MODULARITY ADDRESSES THE LOGICAL DECOMPOSITION OF A PROGRAM. THE Ada FEATURES THAT SUPPORT MODULARITY ARE LISTED IN THE RIGHT HAND COLUMN. A SUBPROGRAM IS A MODULE THAT LOGICALLY PRESENTS A COMPUTATIONAL ACTION. A PACKAGE IS A MODULE CONTAINING SEVERAL MODULES. (LOCALIZATION SUGGESTS THAT ALL MODULES IN A PACKAGE SHOULD BE LOGICALLY RELATED.)
- BRIEFLY EXPLAIN A FEW OF THE RELATIONSHIPS BETWEEN THE CONCEPTS AND THE FEATURES.

SOFTWARE ENGINEERING CONCEPTS APPEARING IN Ada

SOFTWARE ENGINEERING CONCEPTS	Ada FEATURES
MODULARITY	SUBPROGRAMS PACKAGES TASK TYPES SEPARATE COMPILATION
INFORMATION HIDING	SUBPROGRAMS PACKAGES TASK TYPES PRIVATE TYPES
LOCALIZATION	SUBPROGRAMS PACKAGES TASK TYPES SEPARATE COMPILATION
DATA ABSTRACTION	TASK TYPES TYPES PRIVATE TYPES
STRUCTURED PROGRAMMING	SUBPROGRAMS TASK TYPES CONTROL STRUCTURES TYPES

INSTRUCTOR NOTES

- INSTRUCTOR'S TEACHING Ada LANGUAGE MODULES SHOULD USE THIS SLIDE AS A GUIDE FOR RELATING Ada FEATURES TO SOFTWARE ENGINEERING CONCEPTS. IT IS AN IMPORTANT PART OF AN Ada INSTRUCTOR'S JOB TO MAKE THEIR STUDENTS AWARE OF HOW THESE CONCEPTS CAN BE REALIZED IN Ada.

- FOR EXAMPLE, PACKAGES ARE A MAJOR Ada FEATURE. THE SOFTWARE ENGINEERING CONCEPTS THAT PACKAGES SUPPORT ARE LISTED IN THE RIGHT HAND COLUMN. INFORMATION HIDING IS SUPPORTED THROUGH THE PACKAGE SPECIFICATION. THE DETAILS OF SUBPROGRAMS SUPPORTED BY THE INTERFACE ARE HIDDEN IN THE PACKAGE BODY, AS ARE DECLARATIONS DECLARED LOCAL TO THE PACKAGE BODY. LOCALIZATION IS SUPPORTED BY ALLOWING LOGICALLY RELATED ENTITIES TO BE PHYSICALLY GROUPED TOGETHER.

- BRIEFLY EXPLAIN A FEW OF THE RELATIONSHIPS BETWEEN THE CONCEPTS AND THE FEATURES.

Ada SUPPORT OF SOFTWARE ENGINEERING CONCEPTS

Ada FEATURES	SOFTWARE ENGINEERING CONCEPTS
SUBPROGRAMS	MODULARITY INFORMATION HIDING LOCALIZATION STRUCTURED PROGRAMMING
PACKAGES	MODULARITY INFORMATION HIDING LOCALIZATION
TASK TYPES	MODULARITY INFORMATION HIDING LOCALIZATION DATA ABSTRACTION STRUCTURED PROGRAMMING
SEPARATE COMPILATION	MODULARITY LOCALIZATION
TYPES	DATA ABSTRACTION STRUCTURED PROGRAMMING
PRIVATE TYPES	INFORMATION HIDING DATA ABSTRACTION
CONTROL STRUCTURES	STRUCTURED PROGRAMMING

INSTRUCTOR NOTES

- Ada DOES NOT SUPPORT JUST A FEW DESIGN METHODOLOGIES - IT SUPPORTS MOST OF THEM. THIS SLIDE JUST LISTS A FEW OF THE MORE COMMON ONES.

- A COMMON ASPECT OF THE METHODOLOGIES IS THE DECOMPOSITION OF A DESIGN INTO SUCCESSIVELY SMALLER AND SMALLER PIECES. EACH STEP OF THE DECOMPOSITION CAN BE EXPRESSED IN Ada.

- THIRD BULLET

  THIS SHOWS HOW THE BASIC IDEAS OF THE FOUR EXAMPLE DESIGN METHODOLOGIES IS EXPRESSED IN Ada.

Ada SUPPORTS MANY DESIGN METHODOLOGIES

- EXAMPLES
  - OBJECT ORIENTED DESIGN
  - STRUCTURED DESIGN
  - JACKSON DESIGN
  - PROGRAM DESIGN LANGUAGES
- COMMON ASPECT
  - EACH EXPRESSES AN INITIAL DESIGN
  - INITIAL DESIGN DECOMPOSED INTO SMALLER PIECES
  - PROCESS REPEATED ON SMALLER PIECES
- EACH LEVEL IN THE DECOMPOSITION CAN BE EXPRESSED IN Ada
  - OBJECT ORIENTED DESIGN: PRIVATE TYPES AND PACKAGES
  - STRUCTURED DESIGN: PACKAGES
  - JACKSON DESIGN: TASKS
  - PROGRAM DESIGN LANGUAGE: Ada-LIKE

INSTRUCTOR NOTES

- THIS SLIDE IS INCLUDED FOR COMIC RELIEF.
- MAKE SURE THE CLASS <u>COMPOSES</u> ITSELF BEFORE MOVING ON.

AN EXAMPLE OF DECOMPOSITION

A GRAVE ROBBER STARTED DIGGING UP MOZART'S GRAVE. WHEN AT LAST HE CAME TO THE CASKET AND OPENED IT, HE WAS APPALLED TO FIND A FRAIL MAN WITH A 20-FOOT BEARD INSIDE, VERY BUSILY ERASING NOTES FROM SHEETS OF MUSIC. "WHAT ARE YOU DOING?" ASKED THE DUMB-FOUNDED GRAVE ROBBER. "<u>DECOMPOSING</u>," REPLIED MOZART.

INSTRUCTOR NOTES

- ALLOW 90 MINUTES FOR THIS SECTION

SECTION 2

OVERVIEW OF THE CURRICULUM

INSTRUCTOR NOTES

- MOST OF THE REMAINDER OF THIS SECTION IS DEVOTED TO AN OVERVIEW OF EACH MODULE IN THE CURRICULUM
  - SPECIFYING WHAT ARE NOT GOALS IS JUST AS IMPORTANT AS SPECIFYING THE GOALS. THIS HELPS PREVENT AN L201 INSTRUCTOR, FOR EXAMPLE, OF MISTAKENLY TRYING TO TEACH DETAILED Ada, WHICH IS NOT A GOAL OF L201.
  - IN SOME CASES WE ALSO MENTION WHAT IS NOT ASSUMED ABOUT A STUDENT'S BACKGROUND. FOR EXAMPLE, IN L305 WE EXPLICITLY STATE THAT A B.S. IN COMPUTER SCIENCE IS NOT ASSUMED.
  - THE SKETCH OF THE MODULE IS PROVIDED SO THAT INSTRUCTORS IN TRAINING UNDERSTAND WHERE IN THE CURRICULUM THEY CAN FIND MODULES COVERING CERTAIN MATERIAL.

- THE TYPICAL COURSES SUGGESTED FOR THESE FIVE JOB CATEGORIES CONCLUDES THE COURSE.

CURRICULUM OVERVIEW

- EACH MODULE DESCRIPTION INCLUDES
  - GOALS OF THE MODULE
  - ASSUMED BACKGROUND OF STUDENTS
  - SKETCH OF MODULE CONTENTS

- TYPICAL COURSES FOR THESE JOB CATEGORIES WILL BE OUTLINED LATER
  - PROJECT/TASK LEADERS
  - DESIGN CONSULTANTS
  - REAL TIME SYSTEM ARCHITECTS
  - SOFTWARE DESIGNERS
  - PROGRAMMERS

INSTRUCTOR NOTES

- A MIDDLE-LEVEL MANAGER WOULD PROBABLY TAKE L101 AND M101 ONLY.

MANAGEMENT MODULES

- MODULES

  L101 - <u>Ada ORIENTATION FOR MANAGERS</u>

  M101 - <u>SOFTWARE ENGINEERING FOR MANAGERS</u>

  L201 - <u>REAL TIME CONCEPTS</u>

  L303 - <u>Ada FOR SOFTWARE MANAGERS</u>

- INTENDED FOR MANAGERS AND OTHERS WHO DO NOT NEED A DETAILED UNDERSTANDING OF Ada AND SOFTWARE ENGINEERING CONCEPTS

INSTRUCTOR NOTES

- FOR EACH OF THE OVERVIEWS IN THE REST OF THIS SECTION
  - EMPHASIZE WHAT ARE NOT GOALS AND EXPLAIN WHY
  - REMEMBER THAT THE DESCRIPTION OF THE MODULE IS AN OVERVIEW. DETAILED DISCUSSIONS OF THE MODULES ARE GIVEN IN THE CLOSE-UP PARTS OF THIS COURSE.

- FOR L101: REMEMBER THAT THE MANAGERS DO HAVE PROGRAMMING EXPERIENCE, BUT IT MIGHT NOT BE WITH HIGH LEVEL LANGUAGES. DO NOT MAKE THE MISTAKE OF THINKING THEY ARE NOT FAMILIAR WITH PROGRAMMING.

  IT IS IMPORTANT THAT MANAGERS UNDERSTAND WHY Ada CAME INTO EXISTENCE, HOW IT WILL HELP THEM, AND SOME OF THE TRANSITION PROBLEMS THEY MIGHT HAVE.

L101 - Ada ORIENTATION FOR MANAGERS

- GOALS
  - OVERVIEW OF THE DEVELOPMENT OF Ada
  - OVERVIEW OF PROGRAMMING IN Ada
  - OVERVIEW OF Ada FEATURES
  - OVERVIEW OF TRANSITION ISSUES
  - OVERVIEW OF CURRENT STATUS OF THE Ada EFFORT
- GOALS DO <u>NOT</u> INCLUDE
  - TEACH Ada PROGRAMMING
- STUDENT BACKGROUND
  - PROGRAMMING EXPERIENCE (BUT NOT NECESSARILY IN HIGH ORDER LANGUAGES)
- MODULE OVERVIEW (1 DAY)
  - THIS MODULE GIVES AN OVERVIEW OF THE DEVELOPMENT OF Ada, THE Ada LANGUAGE AND THE Ada ENVIRONMENT. THE COURSE EMPHASIZES THE USE OF Ada IN THE TOTAL PROJECT DEVELOPMENT.
  - TOPICS COVERED INCLUDE
    - BACKGROUND AND RATIONALE FOR Ada
    - WHAT IS A HIGH LEVEL LANGUAGE
    - HOW Ada DIFFERS FROM OTHER HIGH LEVEL LANGUAGES
    - WHAT Ada CAN DO FOR AN ORGANIZATION
    - CURRENT STATUS OF Ada
    - WHAT TO EXPECT IN THE FUTURE

INSTRUCTOR NOTES

M101 - SOFTWARE ENGINEERING FOR MANAGERS

- GOALS
  - PROVIDE GENERAL UNDERSTANDING OF SOFTWARE ENGINEERING CONCEPTS
  - ESTABLISH A RELATIONSHIP BETWEEN SOFTWARE ENGINEERING AND Ada
- GOALS DO <u>NOT</u> INCLUDE
  - HOW TO MANAGE SOFTWARE ENGINEERING
- STUDENT BACKGROUND
  - SOME PROGRAMMING EXPERIENCE
  - HOL BACKGROUND <u>NOT</u> ASSUMED
- MODULE OVERVIEW (1 DAY)
  - THIS MODULE DESCRIBES THE SOFTWARE CRISIS, WHAT SOFTWARE ENGINEERING IS, AND HOW IT CAN BE USED TO ALLEVIATE THE SOFTWARE CRISIS. IT ALSO DISCUSSES THE ROLE OF Ada IN SOFTWARE ENGINEERING.
  - TOPICS COVERED INCLUDE
    - LIFE-CYCLE MODEL
    - QUALITY ASSURANCE
    - VERIFICATION AND VALIDATION
    - CONFIGURATION MANAGEMENT
    - SOFTWARE ENGINEERING PRINCIPLES
    - COMPLEXITY MANAGEMENT
    - TOOLS AND METHODS FOR EACH LIFE-CYCLE PHASE

INSTRUCTOR NOTES

- TYPICALLY, THE TYPE OF MANAGER TAKING THIS COURSE WILL BE A PROJECT/TASK LEADER, QA ENGINEER.

- PEOPLE TAKING THIS COURSE DO NOT CARE WHERE THE SEMICOLONS GO, NOR DO THEY CARE ABOUT ALL THE FORMS OF GENERIC PARAMETERS. IT IS IMPORTANT FOR THEM TO KNOW THAT GENERIC PROGRAMMING EXISTS, ITS GENERAL CAPABILITIES, AND WHY IT IS USEFUL.

- PEOPLE TAKING THIS COURSE ARE INTERESTED IN LONG RANGE ASPECTS OF Ada SUCH AS PORTABILITY, READABILITY, ETC.

L201 - Ada FOR SOFTWARE MANAGERS

- GOALS
  - DEVELOP CONCEPTUAL KNOWLEDGE OF ADA
  - RECOGNIZE HIGH/POOR QUALITY DESIGNS AND CODE IN Ada
  - DEVELOP AN UNDERSTANDING OF PORTABILITY AND REUSABILITY ISSUES
- GOALS DO NOT INCLUDE
  - TEACH Ada DESIGN
  - TEACH Ada CODING
- STUDENT BACKGROUND
  - Ada ORIENTATION FOR MANAGERS (L101) OR Ada TECHNICAL OVERVIEW (L102)
  - SOFTWARE ENGINEERING FOR MANAGERS (M101) OR INTRODUCTION TO SOFTWARE ENGINEERING (M102)
- MODULE OVERVIEW (3 DAYS)
  - THIS MODULE PRESENTS THE ENTIRE Ada LANGUAGE FROM THE POINT OF VIEW OF A TECHNICAL MANAGER WHO WILL DIRECT A SOFTWARE PROJECT WITHOUT PERSONALLY PRODUCING DESIGNS OR CODE.
  - USING Ada FEATURES IN DESIGN
    - STRONG TYPING
    - PACKAGES
    - SUBPROGRAMS
    - TASKS
    - GENERICS
    - OVERLOADING
    - EXCEPTIONS
    - LOW-LEVEL FEATURES
  - CHARACTERISTICS OF GOOD Ada DESIGN
    - READABILITY
    - MODULARITY
    - USE OF Ada CONSTRUCTS/LOW-LEVEL FEATURES
    - DESIGN FOR REUSABILITY AND PORTABILITY

INSTRUCTOR NOTES

- MAKE SURE YOU EMPHASIZE THAT A REAL TIME PROGRAMMING BACKGROUND IS NOT REQUIRED.
- TASK/PROJECT LEADERS NEED TO UNDERSTAND CAPABILITIES OF Ada TASKING AND HOW STANDARD DESIGNS (ESPECIALLY REAL TIME DESIGNS) CAN BE IMPLEMENTED IN Ada.

L303 - REAL TIME CONCEPTS

- GOALS
  - GIVE MANAGERS CONCEPTUAL UNDERSTANDING OF APPROACHES TO REAL TIME/CONCURRENT PROGRAMMING IN Ada
  - DEMONSTRATE Ada AS A VIABLE LANGUAGE FOR SOLVING REAL TIME PROBLEMS
  - PREPARE PROJECT LEADERS TO UNDERSTAND DESIGNS AND SETTLE DISPUTES
  - UNDERSTAND PERFORMANCE ISSUES
- GOALS DO NOT INCLUDE
  - ENABLE STUDENTS TO WRITE REAL TIME OR CONCURRENT PROGRAMS
  - TEACH SPECIFIC PERFORMANCE IMPROVEMENT TECHNIQUES IN DETAIL
- STUDENT BACKGROUND
  - Ada FOR SOFTWARE MANAGERS (L201)
  - REAL TIME/CONCURRENT PROGRAMMING BACKGROUND NOT ASSUMED
- MODULE OVERVIEW (1 DAY)
  - THIS MODULE PRESENTS THE TASKING FEATURES OF Ada AT THE CONCEPTUAL LEVEL NECESSARY TO UNDERSTAND REAL TIME/CONCURRENT PROGRAMMING ISSUES IN Ada
  - TOPICS COVERED INCLUDE
    - CONCEPTS OF CONCURRENT PROGRAMMING
      - REASONS FOR AND PROBLEMS IN
      - RUNTIME SYSTEMS
    - Ada TASKING CONCEPTS
      - RENDEZVOUS AND SELECT STATEMENTS
      - ABORTING TASKS, EXCEPTIONS, INTERRUPTS, PRIORITIES
    - FUNDAMENTAL DESIGNS
      - MONITORS, MESSAGE BUFFERS, CYCLIC PROCESSING
    - IMPROVING PERFORMANCE

INSTRUCTOR NOTES

INTRODUCTORY LANGUAGE MODULES

- MODULES
  - L102 - <u>Ada TECHNICAL OVERVIEW</u>
  - L103 - <u>INTRODUCTION TO Ada - A HIGHER ORDER LANGUAGE</u>
  - L202 - <u>BASIC Ada PROGRAMMING</u>
- INTENDED FOR
  - PROGRAMMERS AND OTHERS WHO NEED MORE DETAILED UNDERSTANDING OF Ada
  - ANYONE WHO WILL ATTEND ADVANCED LANGUAGE MODULES

INSTRUCTOR NOTES

- BY GIVING AN OVERVIEW OF Ada CONCEPTS AND FEATURES, THIS COURSE SERVES TWO PURPOSES.

  - GIVES A HIGH-LEVEL OVERVIEW TO THOSE PEOPLE WHO ONLY NEED TO UNDERSTAND Ada CONCEPTS BUT NOT DETAILS.

    (EXAMPLE: LET SUCH A PERSON KNOW THAT A PACKAGE IS NOT SOMETHING DELIVERED BY PARCEL POST.)

  - FOR STUDENTS GOING ON TO L202, THIS SOLVES THE PROBLEM OF USING AN Ada FEATURE BEFORE IT IS DESCRIBED.

    FOR EXAMPLE: IN L202, OBJECT DECLARATIONS CAN BE SHOWN IN THE DECLARATIVE PART OF A SUBPROGRAM BEFORE SUBPROGRAMS ARE DESCRIBED.

L102 - Ada TECHNICAL OVERVIEW

- GOALS
  - INTRODUCE THE STUDENT TO BASIC Ada CONCEPTS
- GOALS DO NOT INCLUDE
  - TEACHING THE DETAILS OF THE LANGUAGE
- STUDENT BACKGROUND
  - PROGRAMMING EXPERIENCE IN HIGHER ORDER LANGUAGES
- MODULE OVERVIEW (1 DAY)
  - THIS MODULE GIVES AN OVERVIEW OF THE DEVELOPMENT OF Ada AND OF THE Ada LANGUAGE
  - TOPICS COVERED INCLUDE
    - BACKGROUND AND RATIONALE FOR ADA
    - DoD LANGUAGE REQUIREMENTS
    - TOP-DOWN VIEW OF Ada
    - LARGE SYSTEM DEVELOPMENT
    - PROGRAM LIBRARY

INSTRUCTOR NOTES

- THIS MODULE CAN BE VIEWED AS L102 FOR STUDENTS WHO HAVE NO HIGHER ORDER LANGUAGE BACKGROUND.

- INSTRUCTORS OF L103 SHOULD NOT MAKE THE MISTAKE OF THINKING THAT THE STUDENTS ARE NOT EXPERIENCED PROGRAMMERS.

L103 - INTRODUCTION TO Ada - A HIGHER ORDER LANGUAGE

- GOALS
  - DESCRIBE HIGHER ORDER LANGUAGES AND WHAT THEY CAN DO
  - ENABLE THE STUDENT TO RECOGNIZE AND UNDERSTAND HOL CONSTRUCTS
  - EXPLAIN HOW PROGRAMMING IN HOL DIFFERS FROM PROGRAMMING IN ASSEMBLER
  - INTRODUCE THE STUDENT TO BASIC Ada CONCEPTS
- GOALS DO <u>NOT</u> INCLUDE
  - TEACHING PROGRAMMING CONCEPTS
  - TEACHING DETAILS OF Ada
- STUDENT BACKGROUND
  - PROGRAMMING EXPERIENCE BUT <u>NOT</u> IN HIGHER ORDER LANGUAGES
- MODULE OVERVIEW (1 DAY)
  - THIS MODULE DESCRIBES HIGHER ORDER LANGUAGE PROGRAMMING TO EXPERIENCED PROGRAMMERS HAVING NO HOL EXPERIENCE. BASIC Ada CONCEPTS ARE DESCRIBED
  - PROS AND CONS OF HOL'S
    - READABILITY
    - PORTABILITY
    - REUSABILITY
    - EFFICIENCY
  - Ada CONCEPTS
    - DATA TYPES
    - CONTROL STRUCTURES
    - SEPARATE COMPILATION
  - OVERVIEW OF IMPLEMENTATION OF HOL FEATURES

INSTRUCTOR NOTES

- EMPHASIZE THAT THE NORMAL AND HIGHLY RECOMMENDED VERSION OF THIS MODULE IS THE 10 DAY VERSION.
- THE 10 DAY VERSION HAS 5 DAYS WORTH OF LAB SPREAD THROUGHOUT THE MODULE.
  - HANDS-ON Ada PROGRAMMING EXPERIENCE
  - HAND-ON ALS EXPERIENCE
- THE 5 DAY VERSION SHOULD BE RESISTED, IF *POSSIBLE*.
- EMPHASIZE THAT SALIENT POINTS OF L102 AND M102 ARE REVIEWED.

L202 - BASIC Ada PROGRAMMING

- GOALS
  - ACQUAINT STUDENTS WITH Ada STYLE OF PROGRAM DEVELOPMENT AND SOFTWARE ENGINEERING METHODS
  - ENABLE STUDENTS TO WRITE MEDIUM-SIZED Ada PROGRAMS
  - OBTAIN READING KNOWLEDGE OF Ada's MORE COMPLEX FEATURES
  - ACQUAINT STUDENTS WITH Ada CODING STYLE CONVENTIONS
- GOALS DO <u>NOT</u> INCLUDE
  - TO TEACH PROGRAMMING
  - TO TEACH SOFTWARE ENGINEERING
  - TO TEACH ALL ASPECTS OF THE Ada LANGUAGE
- STUDENT BACKGROUND
  - PROGRAMMING EXPERIENCE IN HIGH LEVEL LANGUAGES OR <u>INTRODUCTION TO Ada - A HIGHER ORDER LANGUAGE</u> (L103)
  - <u>INTRODUCTION TO SOFTWARE ENGINEERING</u> (M102)
- MODULE OVERVIEW (5 DAYS/10 DAYS)
  - THIS MODULE PRESENTS THE BASIC FEATURES OF THE Ada PROGRAMMING LANGUAGE ALONG WITH GOOD Ada PROGRAMMING STYLE. IT ALSO INCLUDES CONDENSED VERSIONS OF L102 AND M102
  - TOPIC COVERED INCLUDE
    - SCALAR TYPES, ARRAY TYPES, RECORD TYPES, ACCESS TYPES
    - CONTROL STRUCTURES
    - SUBPROGRAMS
    - PACKAGES
    - SEPARATE COMPILATION
    - EXCEPTIONS
    - I/O

VG 931/A

INSTRUCTOR NOTES

ADVANCED LANGUAGE MODULES

- MODULES
  - L305 - ADVANCED Ada TOPICS
  - L401 - REAL TIME SYSTEMS IN Ada
  - L402 - USING THE Ada LANGUAGE REFERENCE MANUAL

- INTENDED FOR
  - DESIGNERS AND OTHERS WHO NEED IN-DEPTH KNOWLEDGE OF ADVANCED Ada FEATURES
  - DESIGN CONSULTANTS AND OTHERS WHO WILL ACT AS LANGUAGE LAWYERS

INSTRUCTOR NOTES

- EMPHASIZE THAT THE <u>NORMAL</u> AND <u>HIGHLY RECOMMENDED</u> VERSION OF THIS MODULE IS THE 10 DAY VERSION.
- THE 10 DAY VERSION HAS 5 DAYS WORTH OF LAB SPREAD THROUGHOUT THE MODULE
  - HANDS-ON Ada PROGRAMMING EXPERIENCE
- THE 5 DAY VERSION SHOULD BE RESISTED, IF POSSIBLE.
- EMPHASIZE THAT B.S. IN COMPUTER SCIENCE NOT REQUIRED, NEITHER IS A STRONG BACKGROUND IN DATA STRUCTURES AND ALGORITHMS.

L305 - ADVANCED Ada TOPICS

- GOALS
  - THOROUGH MASTERY OF MODULARITY AND ENCAPSULATION
  - EXPOSE STUDENTS TO COMPLEX ALGORITHMS AND DATA STRUCTURES
  - TEACH ADA FEATURES RELATED TO ABOVE GOALS
- GOALS DO <u>NOT</u> INCLUDE
  - PROVIDING THOROUGH COVERAGE OF ANY PARTICULAR CLASS OF DATA STRUCTURES AND ALGORITHMS, E.G., SORTING AND SEARCHING.
- STUDENT BACKGROUND
  - <u>BASIC Ada PROGRAMMING</u> (L202)
  - <u>PROGRAMMING METHODOLOGY</u> (M203)
  - STUDENTS ARE <u>NOT</u> ASSUMED TO HAVE B.S. IN COMPUTER SCIENCE OR EQUIVALENT
- MODULE OVERVIEW (5 DAYS/10 DAYS)
  - THIS MODULE TEACHES MODERN ABSTRACTION CONCEPTS AND RELATED FACILITIES OF Ada. IT STRESSES KEY CONCEPTS OF ABSTRACTION AND INFORMATION HIDING IN THE CONTEXT OF ADVANCED PROGRAMMING TECHNIQUES.
  - TOPICS COVERED INCLUDE
    - BASIC DATA STRUCTURE CONCEPTS
      - SETS, LINEAR STRUCTURES : LISTS, STACKS, QUEUES RECURSIVE TYPES, LINKED STRUCTURES : LISTS, STACKS, QUEUES
    - ADVANCED ABSTRACTION FEATURES
      - ENCAPSULATION
      - PRIVATE AND LIMITED PRIVATE TYPE
      - USE OF EXCEPTIONS
      - OVERLOADING
      - GENERIC PROGRAMMING
      - DERIVED TYPES
    - APPLICATIONS
      - TREES, SEARCHING, SORTING, SETS, GRAPHS
    - LOW-LEVEL PROGRAMMING AND LOW-LEVEL I/O
    - OVERVIEW OF Ada TASKING

INSTRUCTOR NOTES

- IMPORTANT TO RECOGNIZE THAT STUDENTS NEED NOT HAVE A BACKGROUND IN REAL TIME/CONCURRENT PROGRAMMING
- MAY HAVE MIXTURE
  - STUDENTS LEARNING ABOUT CONCURRENT PROGRAMMING
  - EXPERIENCED REAL TIME PROGRAMMERS
- ALSO INCLUDED IS 1 DAY OF PROGRAM TUNING
  - MENTIONED THROUGHOUT THE CURRICULUM AS BEING DELAYED UNTIL LATER
  - IT IS NOW LATER!

L401 - REAL TIME SYSTEMS IN Ada

- GOALS
  - TEACH TASKING FEATURES OF Ada
  - INTRODUCE CONCEPTS OF CONCURRENT PROGRAMMING/REAL TIME PROGRAMMING
  - TEACH EXPERIENCED REAL TIME PROGRAMMERS HOW TO USE Ada TASKING FEATURES TO SOLVE PROBLEMS WITH WHICH THEY ARE FAMILIAR
  - TEACH WHEN AND HOW TO IMPROVE PROGRAM PERFORMANCE
- GOALS DO NOT INCLUDE
  - TEACHING ABOUT SPECIFIC IMPLEMENTATIONS OF Ada
  - TEACHING ABOUT SPECIFIC TARGET COMPUTERS
- STUDENT BACKGROUND
  - ADVANCED Ada TOPICS (L305)
  - STUDENTS ARE NOT ASSUMED TO HAVE CONCURRENT PROGRAMMING BACKGROUND

INSTRUCTOR NOTES

L401 - REAL TIME SYSTEMS IN Ada - CONTINUED

- MODULE OVERVIEW (5 DAYS)
  - THIS MODULE TEACHES CONCURRENT PROGRAMMING IN Ada WITH EMPHASIS ON REAL TIME PROGRAMMING. IN ADDITION, THE STUDENT IS TAUGHT HOW AND WHEN TO IMPROVE THE PERFORMANCE OF CONCURRENT/SEQUENTIAL PROGRAMS.
  - TOPICS COVERED INCLUDE
    - CONCEPTS OF CONCURRENT PROGRAMMING
      - REASONS FOR CONCURRENCY
      - PROBLEMS IN CONCURRENT PROGRAMMING
      - RUNTIME SYSTEMS
    - Ada TASKING CONCEPTS
      - RENDEZVOUS
      - SELECT STATEMENTS
      - ABORTING TASKS
      - EXCEPTIONS
      - INTERRUPTS
      - PRIORITIES
    - FUNDAMENTAL TASK DESIGNS
      - MONITORS
      - MESSAGE BUFFERS
      - STREAM-ORIENTED TASK DESIGN
      - CYCLIC PROCESSING
    - IMPROVING PROGRAM PERFORMANCE
      - WHEN TO TUNE
      - SHARED VARIABLES
      - MINIMIZING BLOCKING
      - NON-CONCURRENT TUNING
      - WHAT TO LEAVE TO THE COMPILER

INSTRUCTOR NOTES

- THIS MODULE IS NOT INTENDED FOR MOST PEOPLE. TYPICALLY, PEOPLE TAKING THIS WOULD BE ACTING AS LANGUAGE LAWYERS, I.E., EXPLAINING WHY SOMETHING DOESN'T WORK QUITE THE WAY A PROGRAMMER THINKS IT DOES.

L402 - USING THE Ada LANGUAGE REFERENCE MANUAL

- GOALS
  - DEFINE LANGUAGE TERMS IN THE LRM AND WHERE TERMS ARE DISCUSSED
  - FAMILIARIZE STUDENTS WITH SUBTLE SEMANTIC ISSUES AND HOW TO RESOLVE LANGUAGE ISSUES IN GENERAL.
- GOALS DO <u>NOT</u> INCLUDE
  - TEACH PROGRAMMING
  - TEACH Ada
  - TEACH EVERY DETAIL IN LRM
- STUDENT BACKGROUND
  - <u>ADVANCED Ada TOPICS</u> (L305)
- MODULE OVERVIEW (2 DAYS)
  - THIS MODULE TEACHES THE STUDENT HOW TO USE THE LRM. THE STUDENT WILL UNDERSTAND HOW TO FIND THE SECTIONS OF THE LRM PERTAINING TO A PROBLEM OR QUESTIONS AND HOW TO INTERPRET THESE SECTIONS. KEY CONCEPTS SUCH AS ERRONEOUSNESS AND INCORRECT ORDER DEPENDENCY WILL BE DISCUSSED.
  - TOPICS COVERED INCLUDE
    - PURPOSE OF THE LRM
    - HISTORY OF THE LRM
    - STRUCTURE OF THE LRM
      - SYNTAX NOTATIONS
      - LANGUAGE TERMS
      - ANNEXES AND APPENDICES
    - CHAPTER BY CHAPTER EXAMINATION OF THE LRM

INSTRUCTOR NOTES

METHODOLOGY MODULES

- MODULES
  - M102 - INTRODUCTION TO SOFTWARE ENGINEERING
  - M201 - SOFTWARE ENGINEERING METHODOLOGIES
  - M203 - PROGRAMMING METHODOLOGY
- INTENDED FOR
  - DESIGNERS, PROGRAMMERS AND OTHERS WHO NEED TO UNDERSTAND SOFTWARE ENGINEERING ISSUES AND PROGRAMMING METHODOLOGY.
  - DESIGNERS WHO NEED TO UNDERSTAND SOFTWARE ENGINEERING METHODOLOGIES

INSTRUCTOR NOTES

- THIS IS AN INTRODUCTORY SOFTWARE ENGINEERING COURSE AND MAY WELL BE THE FIRST TIME MANY OF THE STUDENTS HAVE LOOKED AT THEIR ACTIVITY AS PART OF A LARGER PICTURE.

- IT EMPHASIZES THE RELATIONSHIP BETWEEN SOFTWARE ENGINEERING AND Ada.

M102 - INTRODUCTION TO SOFTWARE ENGINEERING

- GOALS
  - DEVELOP CONCEPTUAL UNDERSTANDING OF SOFTWARE ENGINEERING CONCEPTS
  - DEVELOP OVERVIEW UNDERSTANDING OF SOFTWARE ENGINEERING METHODS
  - ESTABLISH RELATIONSHIP BETWEEN SOFTWARE ENGINEERING AND Ada

- GOALS DO NOT INCLUDE
  - TEACH HOW TO USE ANY SPECIFIC DEVELOPMENT METHODOLOGY OR TOOL

- STUDENT BACKGROUND
  - SOME PROGRAMMING EXPERIENCE

- MODULE OVERVIEW (2 DAYS)
  - THIS MODULE TEACHES THE FUNDAMENTAL CONCEPTS OF SOFTWARE ENGINEERING. IT ALSO ATTEMPTS TO MAKE THE STUDENTS AWARE OF WHY SOFTWARE ENGINEERING CONCEPTS ARE BEING TAUGHT WITH Ada.
  - TOPICS COVERED INCLUDE
    - WHAT IS SOFTWARE ENGINEERING
    - PRINCIPLES OF SOFTWARE ENGINEERING
    - SOFTWARE LIFE-CYCLE
    - QUALITY ASSURANCE
    - VERIFICATION AND VALIDATION
    - CONFIGURATION MANAGEMENT
    - TOOLS

INSTRUCTOR NOTES

- NOTE AGAIN THE EMPHASIS ON Ada.
- ORGANIZATIONS MAY WELL BE USING A PARTICULAR METHODOLOGY, SO IT IS NOT REASONABLE TO ENDORSE A PARTICULAR ONE. IN FACT, DOING SO MIGHT ANGER A PARTICULAR ORGANIZATION.

M201 - SOFTWARE ENGINEERING METHODOLOGIES

- GOALS
  - UNDERSTAND GENERAL CONCEPTS BEHIND SEVERAL METHODOLOGIES
  - UNDERSTAND THEIR SCOPE OF APPLICABILITY WITHIN SOFTWARE LIFE-CYCLE
  - UNDERSTAND WHICH METHODS ARE APPROPRIATE IN THE STUDENT'S ORGANIZATION
- GOALS DO NOT INCLUDE
  - ENDORSEMENT OF A PARTICULAR METHODOLOGY
  - FLUENCY IN EVERY METHODOLOGY
  - EXPOSURE TO EVERY EXISTING METHODOLOGY
- STUDENT BACKGROUND
  - SOFTWARE ENGINEERING FOR MANAGERS (M101) OR INTRODUCTION TO SOFTWARE ENGINEERING (M102)
- MODULE OVERVIEW (5 DAYS)
  - THIS MODULE PROVIDES THE STUDENT WITH A THOROUGH UNDERSTANDING OF SOFTWARE METHODOLOGIES AND HOW THEY MAY BE USED WITH Ada
  - TOPICS COVERED INCLUDE

● SADT	● WARNIER-ORR
● SREM	● HIPO
● ENTITY DIAGRAMS	● HOS
● BACHMAN DIAGRAMS	● STRUCTURED PROGRAMMING
● PARNAS AND OBJECTED-ORIENTED DESIGN	● DESIGN HEURISTICS
● CONSTANTINE-MYERS STRUCTURED DESIGN	● PROGRAM CORRECTNESS
● JACKSON DESIGN	

VG 931/A

INSTRUCTOR NOTES

- THIS IS A PARTICULARLY IMPORTANT COURSE FOR ANYONE WHO WILL BE DOING SIGNIFICANT CODING IN Ada OR CODE REVIEWING OF Ada SOURCE. IT DISCUSSES HOW TO USE Ada EFFECTIVELY AS A PROGRAMMING TOOL.

M203 - PROGRAMMING METHODOLOGY

- GOALS
  - TEACH MODERN CODING TECHNIQUES APPLICABLE TO Ada
  - PROVIDE TECHNICAL BACKGROUND NECESSARY TO APPLY THE TECHNIQUES
- GOALS DO NOT INCLUDE
  - TEACH THE Ada LANGUAGE
- STUDENT BACKGROUND
  - Ada ORIENTATION FOR MANAGERS (L101) OR Ada TECHNICAL OVERVIEW (L102) OR INTRODUCTION TO Ada - A HIGHER ORDER LANGUAGE (L103)
- MODULE OVERVIEW (1 1/2 DAYS)
  - THIS MODULE TEACHES CODING AND DOCUMENTATION CONVENTIONS, STRUCTURED PROGRAMMING AND PROGRAMMING STYLE
  - TOPICS COVERED INCLUDE
    - STRUCTURED PROGRAMMING CONCEPTS
    - BASIC CONTROL STRUCTURES
    - DOCUMENTATION
    - PROGRAMMING STYLE
    - STEPWISE REFINEMENT
    - ENSURING RELIABILITY

INSTRUCTOR NOTES

ENVIRONMENT MODULES

- MODULES
  - E100 - Ada LANGUAGE SYSTEM (ALS) USER COURSE
  - E200 - Ada LANGUAGE SYSTEM (ALS) ADMINISTRATOR COURSE

- INTENDED FOR
  - USER'S OF THE ALS
  - ALS SYSTEM ADMINISTRATORS

INSTRUCTOR NOTES

- REMEMBER THAT Ada HAS AN ENVIRONMENT TO MAKE A PROJECT MORE EFFICIENT AND ORGANIZED.
- ALLOWS STUDENTS TO LEARN ONE ENVIRONMENT.
- THIS MODULE TEACHES THE STUDENTS HOW TO USE ALS AND USE IT EFFECTIVELY.

E100 - Ada LANGUAGE SYSTEM (ALS) USER COURSE

- GOALS
  - LEARN HOW TO USE ALL THE ALS TOOLS
  - LEARN THE FEATURES OF THE ALS DATABASE
  - LEARN AND GAIN EXPERIENCE WITH THE ALS COMMAND LANGUAGE
  - LEARN HOW THE ALS SUPPORTS CONFIGURATION MANAGEMENT

- GOALS DO <u>NOT</u> INCLUDE
  - TEACHING THE Ada LANGUAGE
  - TEACHING VAX/VMS EDITOR

- STUDENT BACKGROUND
  - SOME PROGRAMMING EXPERIENCE
  - KNOWLEDGE OF VAX/VMS EDITOR (EDT)

- MODULE OVERVIEW (10 DAYS)
  - THIS MODULE TEACHES THE STUDENT HOW TO USE THE ALS. THE STUDENT WILL LEARN HOW TO USE THE ALS TOOLS TO COMPILE, LINK AND EXECUTE Ada PROGRAMS AND HOW TO CREATE NEW TOOLS.
  - TOPICS COVERED INCLUDE

● ALS OVERVIEW	● LINKING Ada PROGRAMS
● ENVIRONMENT DATABASE	● DEBUGGING Ada PROGRAMS
● COMMAND LANGUAGE	● ASSEMBLING
● FILE SYSTEM	● WRITING TOOLS IN Ada
● COMPILING Ada PROGRAMS	● CONFIGURATION MANAGEMENT

INSTRUCTOR NOTES

- NOT ONLY DOES THIS TEACH THE STUDENTS HOW TO ADMINISTER THE ALS, IT ALSO TEACHES THEM HOW TO ADMINISTER IT EFFECTIVELY.

E200 - Ada LANGUAGE SYSTEM (ALS) USER COURSE

- GOALS
  - HOW TO INSTALL AN ALS
  - HOW TO ADMINISTER AN ALS

- GOALS DO NOT INCLUDE
  - TEACHING THE ALS
  - TEACHING VAX/VMS

- STUDENT BACKGROUND
  - Ada LANGUAGE SYSTEM (ALS) USER COURSE (E100)
  - HANDS-ON EXPERIENCE WITH A VAX COMPUTER AND VAX/VMS OPERATING SYSTEM

- MODULE OVERVIEW (3 DAYS)
  - THIS MODULE PREPARES THE STUDENT TO INSTALL AND ADMINISTER AN ALS. THE STUDENT IS TAUGHT HOW TO AUTHORIZE ALS USERS AND PERFORM SYSTEM BACKUPS
  - TOPICS COVERED INCLUDE
    - INSTALLING THE ALS
    - USER AND TEAM MANAGEMENT
    - BACKUP AND ARCHIVING
    - INCREMENTAL UPDATES

INSTRUCTOR NOTES

- THIS SLIDE SUGGESTS SOME GUIDELINES FOR PACKAGING MODULES.

- THE VIEWPOINT DETERMINES WHETHER THE EMPHASIS IS ON DETAILS OR CONCEPTS. IN THE MANAGER'S VIEWPOINT WE STEP AWAY FROM DETAILS AND CONCENTRATE ON CONCEPTS.

  EXAMPLES: FROM L303

  - DESCRIBES TASK ACTIVATION AND TERMINATION WITHOUT GOING INTO DETAILS OF ARRAYS OF TASKS, TASKS AS RECORD COMPONENTS, OR MASTERS.
  - DESCRIBES WHEN AND HOW THE ABORT STATEMENT SHOULD BE USED WITHOUT DISCUSSING ABNORMAL TASKS
  - DESCRIBES STREAM-ORIENTED TASK DESIGN WITHOUT GOING INTO DETAILED EXAMINATION OF CODE.

SUGGESTIONS FOR PACKING MODULES INTO COURSES

- DEFINE THE VIEWPOINT
  - PRACTITIONER'S VIEWPOINT: AIMED AT PEOPLE WHO WILL WRITE Ada CODE
  - MANAGER'S VIEWPOINT: APPLIES TO ANYONE NOT NEEDING WORKING KNOWLEDGE OF Ada
    - SHORTER AND CONCEPT-ORIENTED
    - NOT SUPERFICIAL - EMPHASIS ON CONCEPTS RATHER THAN DETAILS MAY MAKE COURSE DEEPER
    - WELL SUITED FOR CONTRACT MONITORS AND PEOPLE DOING IN-DEPTH QA
- DEFINE THE LEVEL
  - PRACTITIONER'S VIEWPOINT
    - 100 SERIES ARE INTRODUCTORY COURSES, INTENDED AS PREREQUISITES
    - HIGHER SERIES INDICATE MORE AND MORE ADVANCED MODULES
  - MANAGER VIEWPOINT
    - 100 SERIES PROVIDES HIGH-LEVEL OVERVIEW FOR TOP MANAGEMENT
    - HIGHER SERIES APPROPRIATE FOR SOFTWARE MANAGERS, QA, ANALYSTS, ETC.
- SELECT MAIN MODULES FOR COURSE

  EXAMPLE: L202, L305, L401 FOR REAL TIME SYSTEM ARCHITECTS
- SEARCH FOR RELATED COURSES
  - METHODOLOGY AND/OR ENVIRONMENT
  - TEACH IN PARALLEL WITH LANGUAGE COURSES IF POSSIBLE

INSTRUCTOR NOTES

- THIS SLIDE AND THE NEXT FOUR SHOW TYPICAL COURSES FOR THE JOB LEVELS DESCRIBED IN SECTION 1.

- WHEN PRESENTING THESE SLIDES EXPLAIN WHY THE COURSE CONTAINS THE MODULES.

- THIS AUDIENCE NEEDS TO HAVE CONCEPTUAL UNDERSTANDING OF FULL Ada. THEREFORE PROJECT/TASK LEADERS NEED

  L101

  L201

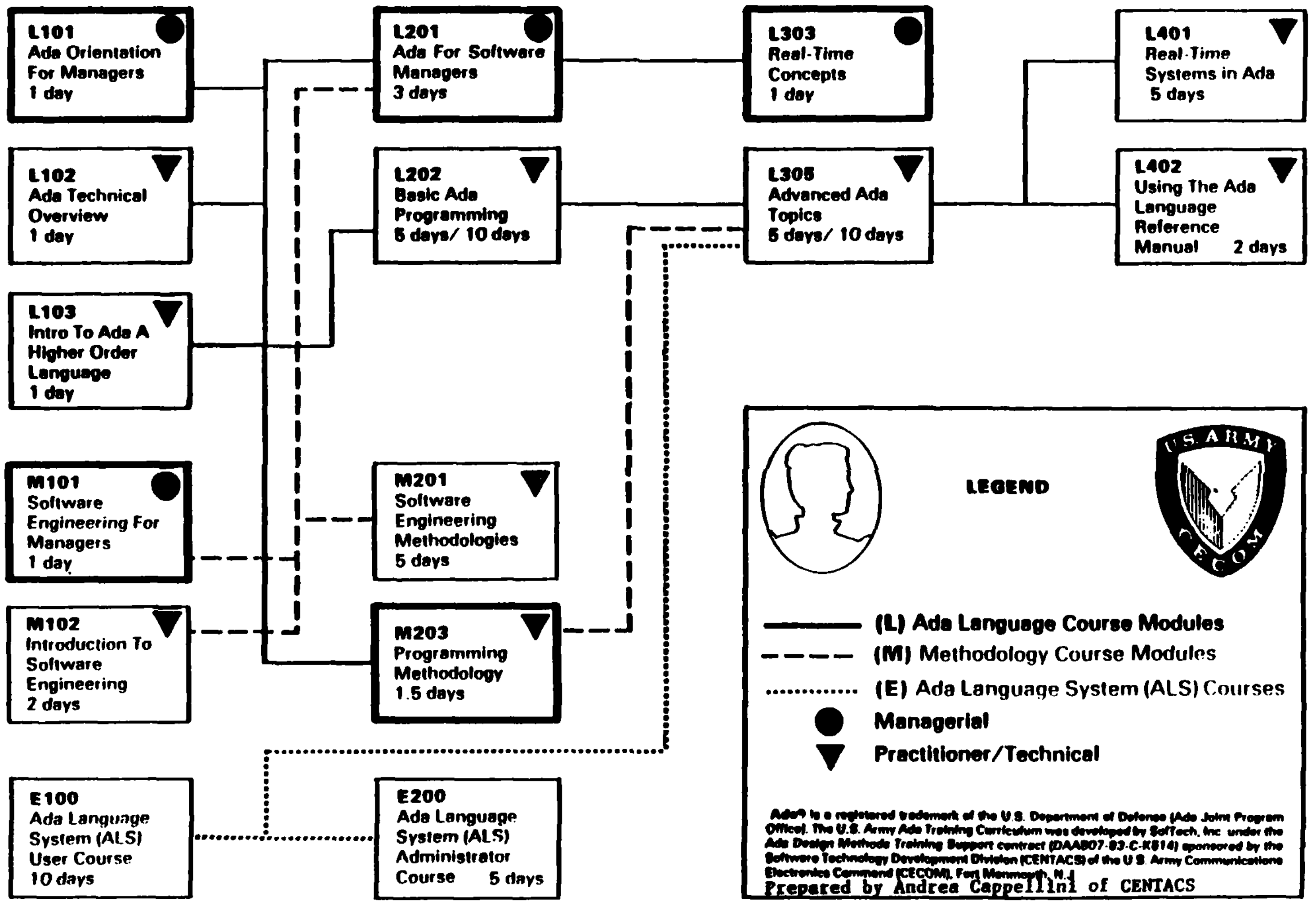
  L303

  THEY ALSO NEED TO UNDERSTAND SOFTWARE ENGINEERING AND PROGRAMMING METHODOLOGY. THEREFORE THEY NEED

  M101

  M203

SUGGESTED COURSE FOR PROJECT/TASK LEADERS


L101 Ada Orientation For Managers 1 day ●
L102 Ada Technical Overview 1 day ▼
L103 Intro To Ada A Higher Order Language 1 day ▼
M101 Software Engineering For Managers 1 day ●
M102 Introduction To Software Engineering 2 days ▼
E100 Ada Language System (ALS) User Course 10 days
L201 Ada For Software Managers 3 days ●
L202 Basic Ada Programming 5 days/ 10 days ▼
M201 Software Engineering Methodologies 5 days ▼
M203 Programming Methodology 1.5 days ▼
E200 Ada Language System (ALS) Administrator Course 5 days
L303 Real-Time Concepts 1 day ●
L305 Advanced Ada Topics 5 days/ 10 days ▼
L401 Real-Time Systems in Ada 5 days ▼
L402 Using The Ada Language Reference Manual 2 days ▼
LEGEND
U.S. ARMY
CECOM
——— (L) Ada Language Course Modules
– – – – (M) Methodology Course Modules
.... (E) Ada Language System (ALS) Courses
● Managerial
▼ Practitioner/Technical

Ada® is a registered trademark of the U.S. Department of Defense (Ada Joint Program Office). The U.S. Army Ada Training Curriculum was developed by SofTech, Inc under the Ada Design Methods Training Support contract (DAAB07-83-C-K514) sponsored by the Software Technology Development Division (CENTACS) of the U.S. Army Communications Electronics Command (CECOM), Fort Monmouth, N.J.

Prepared by Andrea Cappellini of CENTACS

INSTRUCTOR NOTES

- THIS AUDIENCE NEEDS TO UNDERSTAND THE FULL Ada LANGUAGE AND MUST ALSO BE ABLE TO RESOLVE Ada ISSUES. THEREFORE DESIGN CONSULTANTS NEED

  L102

  L202

  L305

  L401

  L402

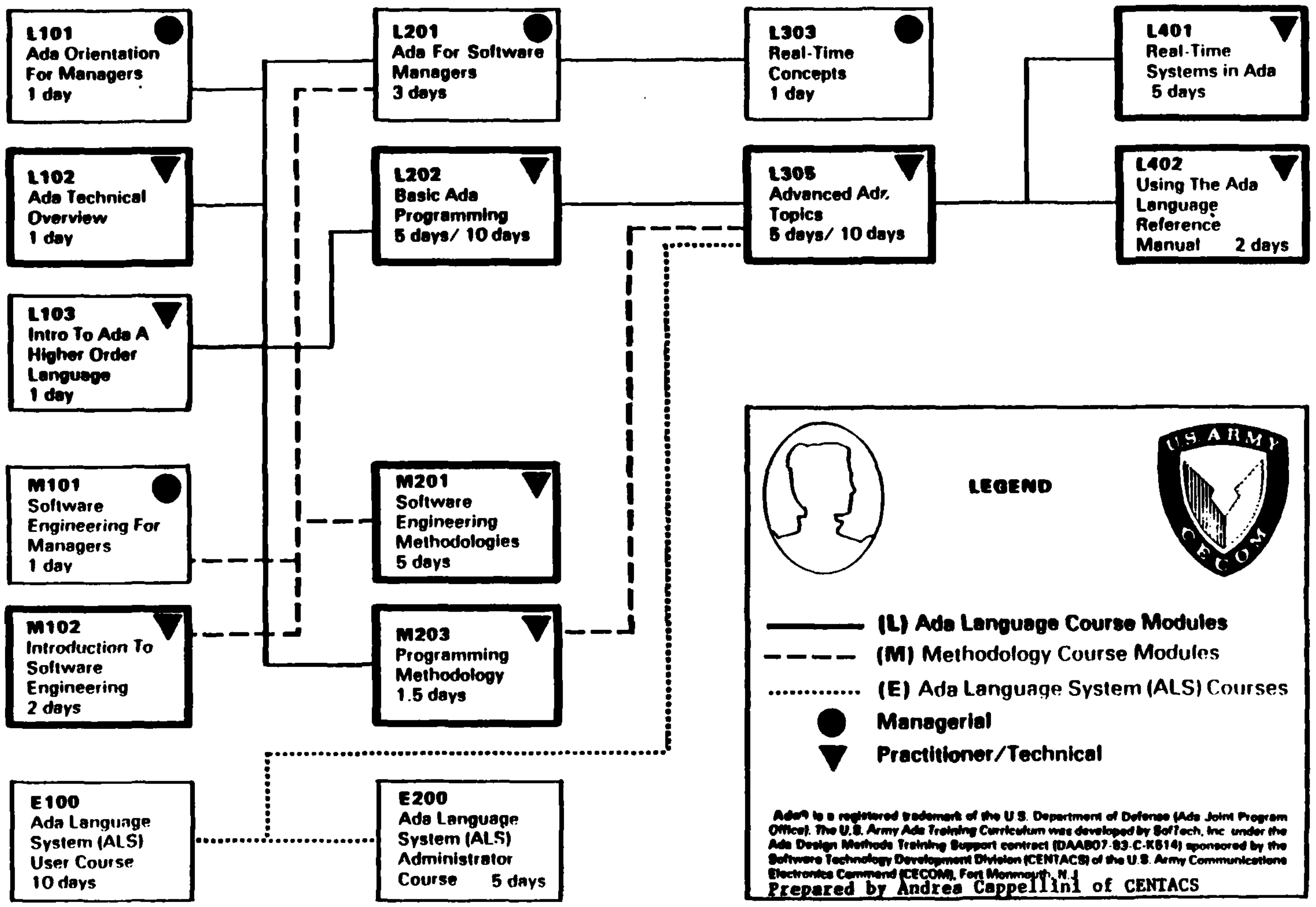
  SINCE THEY ALSO NEED TO UNDERSTAND SOFTWARE ENGINEERING METHODOLOGIES AND PROGRAMMING METHODOLOGY, THEY NEED

  M102

  M201

  M203

SUGGESTED COURSE FOR DESIGN CONSULTANTS


L101 Ada Orientation For Managers 1 day
L102 Ada Technical Overview 1 day
L103 Intro To Ada A Higher Order Language 1 day
M101 Software Engineering For Managers 1 day
M102 Introduction To Software Engineering 2 days
E100 Ada Language System (ALS) User Course 10 days
L201 Ada For Software Managers 3 days
L202 Basic Ada Programming 5 days/ 10 days
M201 Software Engineering Methodologies 5 days
M203 Programming Methodology 1.5 days
E200 Ada Language System (ALS) Administrator Course 5 days
L303 Real-Time Concepts 1 day
L305 Advanced Ada Topics 5 days/ 10 days
L401 Real-Time Systems in Ada 5 days
L402 Using The Ada Language Reference Manual 2 days
LEGEND
US ARMY
CECOM
(L) Ada Language Course Modules
(M) Methodology Course Modules
(E) Ada Language System (ALS) Courses
Managerial
Practitioner/Technical

Ada® is a registered trademark of the U.S. Department of Defense (Ada Joint Program Office). The U.S. Army Ada Training Curriculum was developed by SofTech, Inc under the Ada Design Methods Training Support contract (DAAB07-83-C-K514) sponsored by the Software Technology Development Division (CENTACS) of the U.S. Army Communications Electronics Command (CECOM), Fort Monmouth, N.J.

Prepared by Andrea Cappellini of CENTACS

INSTRUCTOR NOTES

- THIS AUDIENCE NEEDS TO UNDERSTAND FULL Ada. THEREFORE REAL TIME SYSTEM ARCHITECTS NEED

  L102

  L202

  L305

  L401

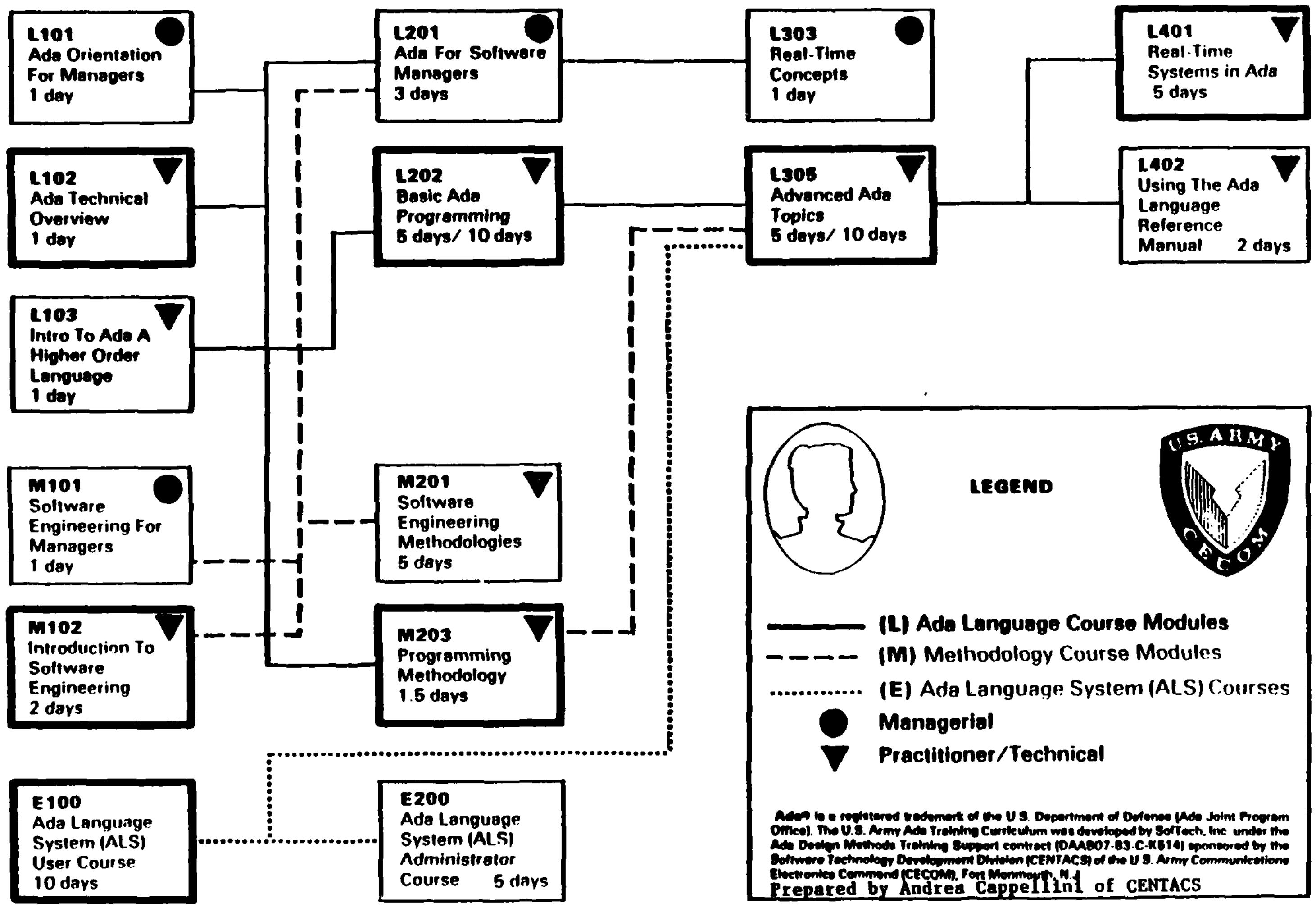
  THEY ALSO NEED TO UNDERSTAND SOFTWARE ENGINEERING, PROGRAMMING METHODOLOGIES AND HOW TO USE THE ALS. THEREFORE THEY NEED

  M102

  M203

  E100

SUGGESTED COURSE FOR REAL-TIME SYSTEM ARCHITECTS


L101 Ada Orientation For Managers 1 day ●
L102 Ada Technical Overview 1 day ▼
L103 Intro To Ada A Higher Order Language 1 day ▼
M101 Software Engineering For Managers 1 day ●
M102 Introduction To Software Engineering 2 days ▼
E100 Ada Language System (ALS) User Course 10 days
L201 Ada For Software Managers 3 days ●
L202 Basic Ada Programming 5 days/ 10 days ▼
M201 Software Engineering Methodologies 5 days ▼
M203 Programming Methodology 1.5 days ▼
E200 Ada Language System (ALS) Administrator Course 5 days
L303 Real-Time Concepts 1 day ●
L305 Advanced Ada Topics 5 days/ 10 days ▼
L401 Real-Time Systems in Ada 5 days ▼
L402 Using The Ada Language Reference Manual 2 days ▼
LEGEND
U.S. ARMY
CECOM
(L) Ada Language Course Modules
(M) Methodology Course Modules
(E) Ada Language System (ALS) Courses
● Managerial
▼ Practitioner/Technical

Ada® is a registered trademark of the U.S. Department of Defense (Ada Joint Program Office). The U.S. Army Ada Training Curriculum was developed by SofTech, Inc under the Ada Design Methods Training Support contract (DAAB07-83-C-K514) sponsored by the Software Technology Development Division (CENTACS) of the U.S. Army Communications Electronics Command (CECOM), Fort Monmouth, N.J.

Prepared by Andrea Cappellini of CENTACS

INSTRUCTOR NOTES

- THIS AUDIENCE NEEDS TO UNDERSTAND ADVANCED Ada. THEREFORE SOFTWARE DESIGNERS NEED

  L102

  L202

  L305

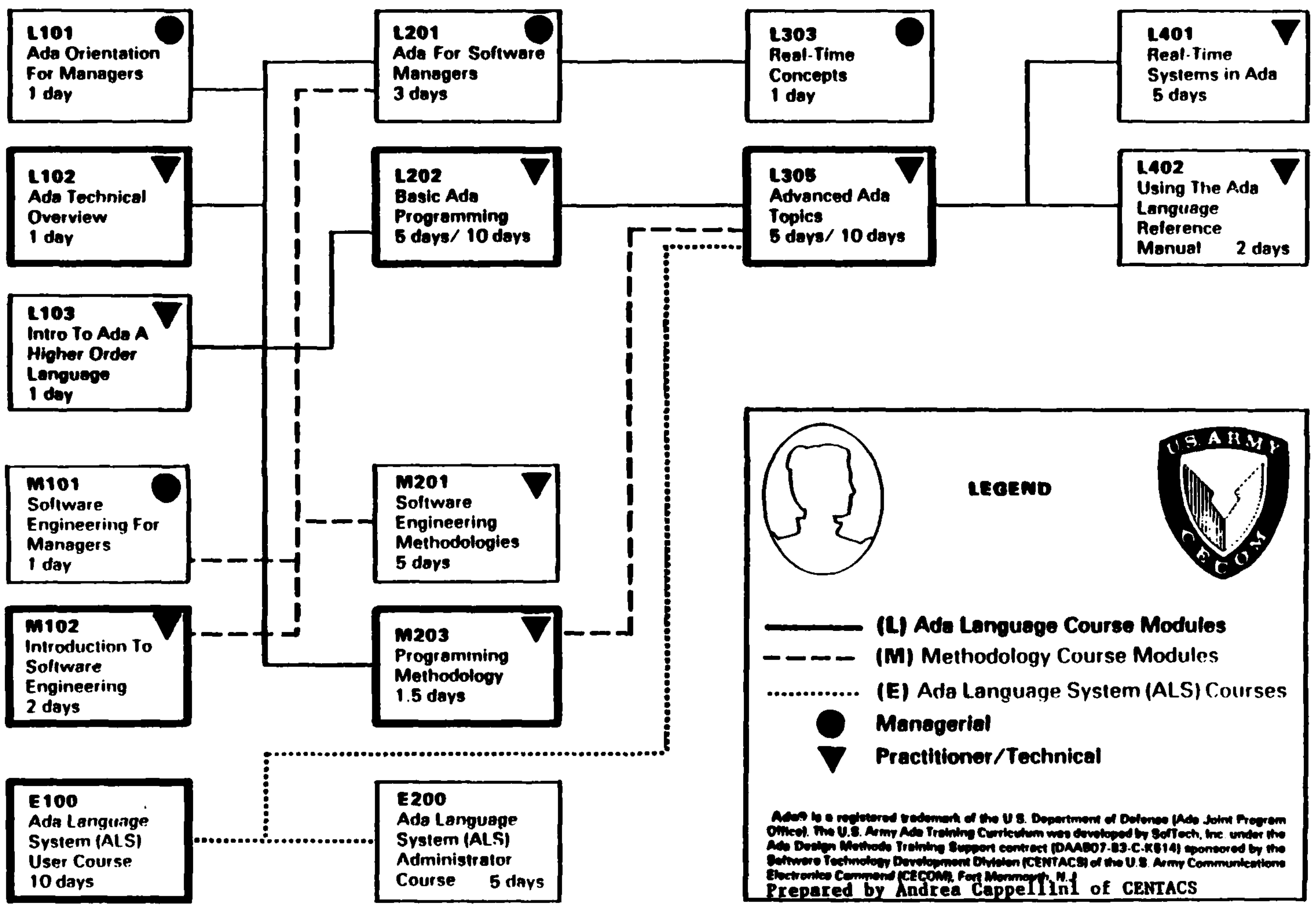
  THEY ALSO NEED TO UNDERSTAND SOFTWARE ENGINEERING, PROGRAMMING METHODOLOGY AND HOW TO USE THE ALS. THEREFORE THEY NEED

  M102

  M203

  E100

SUGGESTED COURSE FOR SOFTWARE DESIGNERS


L101 Ada Orientation For Managers 1 day
L102 Ada Technical Overview 1 day
L103 Intro To Ada A Higher Order Language 1 day
M101 Software Engineering For Managers 1 day
M102 Introduction To Software Engineering 2 days
E100 Ada Language System (ALS) User Course 10 days
L201 Ada For Software Managers 3 days
L202 Basic Ada Programming 5 days/ 10 days
M201 Software Engineering Methodologies 5 days
M203 Programming Methodology 1.5 days
E200 Ada Language System (ALS) Administrator Course 5 days
L303 Real-Time Concepts 1 day
L305 Advanced Ada Topics 5 days/ 10 days
L401 Real-Time Systems in Ada 5 days
L402 Using The Ada Language Reference Manual 2 days
LEGEND
U.S. ARMY
CECOM
(L) Ada Language Course Modules
(M) Methodology Course Modules
(E) Ada Language System (ALS) Courses
Managerial
Practitioner/Technical

Ada® is a registered trademark of the U.S. Department of Defense (Ada Joint Program Office). The U.S. Army Ada Training Curriculum was developed by SofTech, Inc. under the Ada Design Methods Training Support contract (DAAB07-83-C-K514) sponsored by the Software Technology Development Division (CENTACS) of the U.S. Army Communications Electronics Command (CECOM), Fort Monmouth, N.J.

Prepared by Andrea Cappellini of CENTACS

INSTRUCTOR NOTES

- THIS AUDIENCE NEEDS TO UNDERSTAND BASIC Ada. THEREFORE, PROGRAMMERS NEED

  L102

  L202

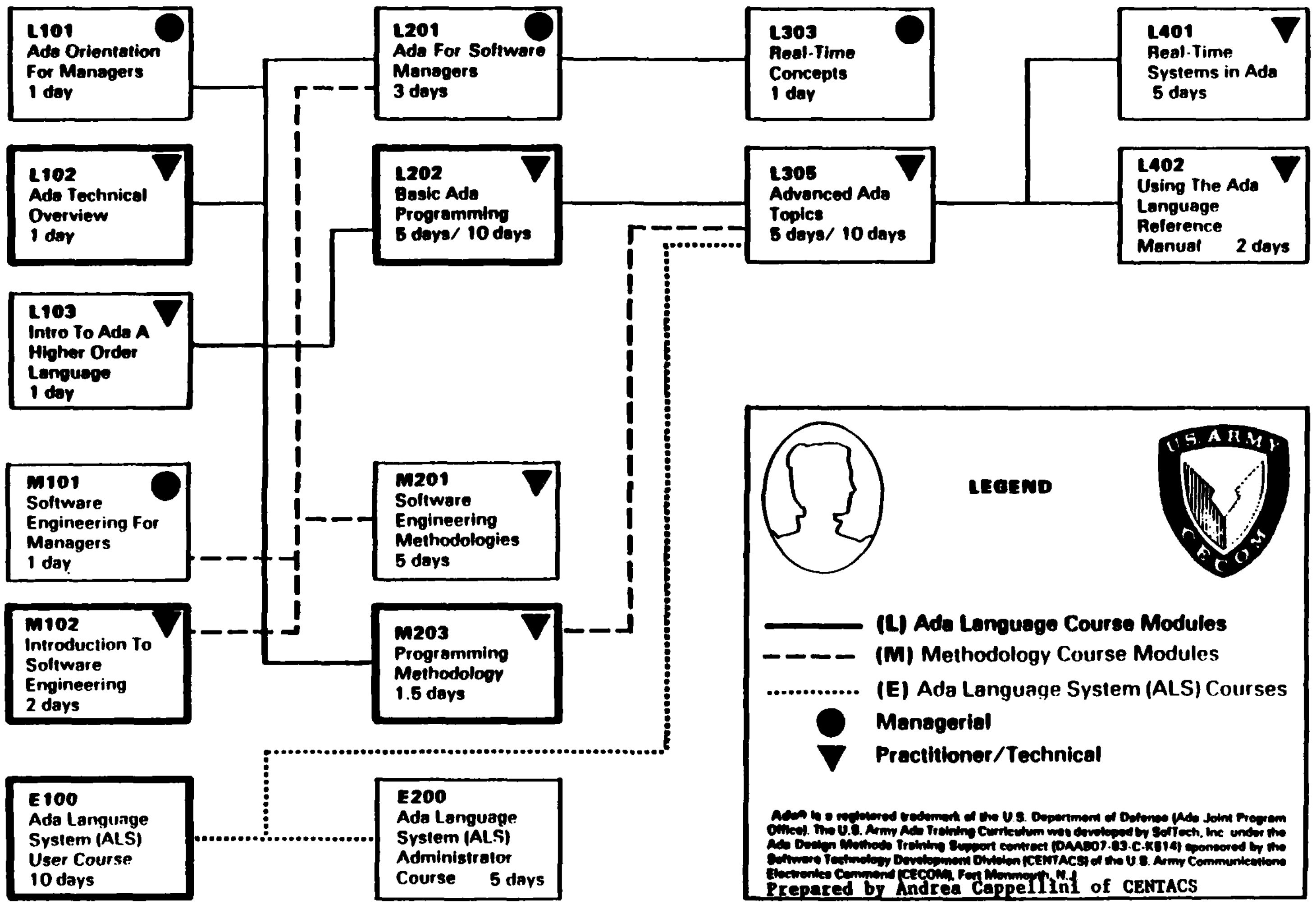
  THEY ALSO NEED TO UNDERSTAND SOFTWARE ENGINEERING, PROGRAMMING METHODOLOGY, AND HOW TO USE THE ALS. THEREFORE, THEY NEED

  M102

  M203

  E100

SUGGESTED COURSE FOR PROGRAMMERS


L101 Ada Orientation For Managers 1 day
L102 Ada Technical Overview 1 day
L103 Intro To Ada A Higher Order Language 1 day
M101 Software Engineering For Managers 1 day
M102 Introduction To Software Engineering 2 days
E100 Ada Language System (ALS) User Course 10 days
L201 Ada For Software Managers 3 days
L202 Basic Ada Programming 5 days/ 10 days
M201 Software Engineering Methodologies 5 days
M203 Programming Methodology 1.5 days
E200 Ada Language System (ALS) Administrator Course 5 days
L303 Real-Time Concepts 1 day
L305 Advanced Ada Topics 5 days/ 10 days
L401 Real-Time Systems in Ada 5 days
L402 Using The Ada Language Reference Manual 2 days
LEGEND
US ARMY
CECOM
(L) Ada Language Course Modules
(M) Methodology Course Modules
(E) Ada Language System (ALS) Courses
Managerial
Practitioner/Technical

Ada® is a registered trademark of the U.S. Department of Defense (Ada Joint Program Office). The U.S. Army Ada Training Curriculum was developed by SofTech, Inc under the Ada Design Methods Training Support contract (DAAB07-83-C-K514) sponsored by the Software Technology Development Division (CENTACS) of the U.S. Army Communications Electronics Command (CECOM), Fort Monmouth, N.J.

Prepared by Andrea Cappellini of CENTACS

INSTRUCTOR NOTES

- ALLOW 45 MINUTES FOR THIS SECTION

SECTION 3

GENERAL CONSIDERATIONS

INSTRUCTOR NOTES

- THESE ARE THE FIVE TOPICS COVERED IN THIS SECTION.
- FOR PREDICTABLE OBJECTIONS, BOTH THE PRACTITIONER'S AND MANAGER'S VIEWPOINTS ARE CONSIDERED.

SPECIAL CONSIDERATIONS FOR THE Ada CURRICULUM

- EXPLAINING WHAT IS DIFFERENT ABOUT Ada
- PREDICTABLE OBJECTIONS
- PREDICTABLE DIFFICULTIES
- TRAPS TO AVOID
- EXERCISES

INSTRUCTOR NOTES

- THIS SLIDE PRESENTS A GOOD HIGH LEVEL RESPONSE TO THE QUESTION "WHAT IS DIFFERENT ABOUT Ada?"
  - THIS MIGHT BE ASKED IN THE LOWER LEVEL LANGUAGE MODULES OR IN THE METHODOLOGY MODULES.
  - THE INSTRUCTOR MIGHT WANT TO ASK THE QUESTION IN CLASS AND THEN ANSWER IT HERSELF/HIMSELF.

- PERFORMANCE IS ADDRESSED AGAIN LATER *IN THIS SECTION*.

EXPLAINING WHAT IS DIFFERENT ABOUT Ada

- THINK MUCH MORE, DEBUG MUCH LESS
  - DESIGN EXPRESSED EASILY IN CODE AT MORE ABSTRACT LEVEL
  - MORE ERRORS DETECTED BY COMPILER THAN TESTING

- SOFTWARE "TALKS" ABOUT THE PROBLEM NOT ABOUT THE MACHINE:

```
COMPUTE_TRAJECTORY;
        NOT
 LOAD ACCUMULATOR
   ROTATE LEFT
   XOR WITH MASK
```

  - HIGHER LEVEL ABSTRACTION
  - TRANSPORTABLE

- MACHINE PERFORMANCE IS NOT ALWAYS THE ONLY PARAMETER TO TUNE
  - SCHEDULE/DEVELOPMENT COST
  - RESPONSIVENESS TO CHANGE
  - RELIABILITY

  ARE AT LEAST AS IMPORTANT AND ARE NOT INCOMPATIBLE

INSTRUCTOR NOTES

- THIS IS THE FIRST OF TWO SLIDES DISCUSSING PROBABLE RESISTANCE FROM THE PRACTITIONER'S VIEWPOINT.

- WHENEVER A PROGRAMMER MOVES FROM A LANGUAGE THEY ARE COMFORTABLE WITH TO A NEW LANGUAGE, IT TAKES A WHILE FOR THE PROGRAMMER TO LEARN HOW TO USE THE NEW LANGUAGE PROPERLY.
  - THE Ada CURRICULUM MINIMIZES THIS PERIOD OF TIME BY TEACHING HOW TO USE Ada TO *SOLVE TYPICAL PROBLEMS*.
  - HABITS DEVELOPED WHILE USING OTHER LANGUAGES MAY HAVE TO BE BROKEN: FOR EXAMPLE, PROGRAMMERS WITH EXPERIENCE IN THE C PROGRAMMING LANGUAGE WILL NEED TO STOP THINKING OF ARRAYS AND POINTERS AS BEING RELATED.

PROBABLE RESISTANCE TO USING Ada: PRACTITIONER'S VIEWPOINT

"I CAN'T FIGURE OUT HOW TO DO IT IN Ada"

- REASONS
  - LACK OF KNOWLEDGE OF ENTIRE Ada LANGUAGE
  - LACK OF UNDERSTANDING OF HOW TO PROGRAM IN Ada
  - TRYING TO PROGRAM IN Ada AS ONE WOULD IN FORTRAN, CMS2, C, JOVIAL, ETC.

INSTRUCTOR NOTES

- FIRST STATEMENT
  - PEOPLE DO RESIST CHANGE, SO THIS SHOULDN'T BE UNEXPECTED. THIS MAY BE ESPECIALLY TRUE FOR PEOPLE WHO TEND TO WORK ALONE (LEARNING Ada WOULD REQUIRE WORKING WITH OTHERS) OR PEOPLE WHO ARE EXPERTS WITH THE CURRENT LANGUAGE USED (AND WOULD SUFFER A LOSS OF PRESTIGE IF THEY WERE REDUCED TO EVERYONE ELSE'S LEVEL OF UNDERSTANDING OF ADA).
  - ALSO, Ada HAS ERRONEOUSLY ACQUIRED THE REPUTATION OF BEING EXTREMELY DIFFICULT TO UNDERSTAND.

- SECOND STATEMENT
  - PEOPLE TEND TO IDENTIFY AN IMPLEMENTATION OF A LANGUAGE WITH A LANGUAGE.
  - MAY BE TRYING TO DO REAL TIME PROGRAMMING WITH AN IMPLEMENTATION WHOSE RUNTIME SYSTEM IS NOT DESIGNED FOR IT.
  - MAY BE USING AN IMPLEMENTATION THAT PROVIDES MINIMUM REQUIRED LOW-LEVEL/IMPLEMENTATION DEPENDENT FEATURES, I.E., NO CODE PROCEDURES, NO INTERFACE TO ASSEMBLER LANGUAGES, NO ADDRESS CLAUSES, ETC.

- THIRD STATEMENT
  - THREE OF MANY POSSIBLE REASONS ARE LISTED.
  - EFFICIENCY WILL BE DISCUSSED LATER IN THIS SECTION.

PROBABLE RESISTANCE TO USING Ada: PRACTITIONER'S VIEWPOINT - CONTINUED

"I DON'T UNDERSTAND IT, AND I WON'T LET YOU FORCE ME TO USE IT"

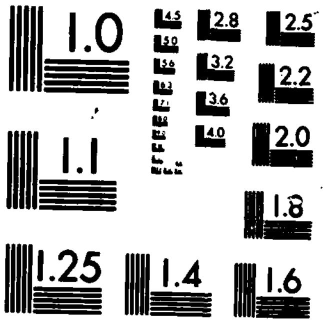
- REASONS:
  - COMFORTABLE DOING BUSINESS AS USUAL INTIMIDATED BY Ada

"THE IMPLEMENTATION IS POORLY MATCHED TO OUR NEEDS."

- REASONS:
  - INAPPROPRIATE RUNTIME SYSTEM
  - IMPLEMENTATION DOES NOT PROVIDE APPROPRIATE LOW-LEVEL FEATURES

"WE TRIED IT WITH A DIFFERENT LANGUACE/COMPILER/CPU AND IT WAS INEFFICIENT."

- REASONS:
  - PREFERENCE FOR ASSEMBLER LANGUAGE
  - INAPPROPRIATE RUNTIME SYSTEM
  - LACK OF UNDERSTANDING OF HOW TO TUNE A PROGRAM



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

INSTRUCTOR NOTES

- THIS IS THE FIRST OF TWO SLIDES DISCUSSING PROBABLE RESISTANCE FROM THE MANAGER'S VIEWPOINT.

- MANY MANAGERS ARE USED TO PROGRAMMERS PICKING UP A BOOK ON C, PASCAL, FORTRAN, ETC., AND LEARNING THE LANGUAGE FROM IT. IT IS MUCH HARDER TO DO WITH Ada, SO MANAGERS MAY BALK AT SPENDING MONEY FOR SEVERAL WEEKS WORTH OF TRAINING AND THEN WAITING FOR THE PROGRAMMERS TO BE EXPERIENCED Ada PROGRAMMERS.

- MANAGERS MUST BE MADE TO REALIZE THAT Ada CURRICULUM REDUCES THE LEARNING CURVE AND THAT BY THE END OF A PROJECT, THE RESULTING HIGHER QUALITY SOFTWARE WILL HAVE COST LESS TO PRODUCE.

PROBABLE RESISTANCE TO USING Ada: MANAGER'S VIEWPOINT

"IT WILL TAKE TOO LONG FOR MY PEOPLE TO BE PRODUCTIVE Ada PROGRAMMERS"

- REASONS:
  - USED TO PROGRAMMERS PICKING UP LANGUAGE ON OWN
  - DON'T REALIZE CURRICULUM TEACHES HOW TO USE Ada AS WELL AS LANGUAGE FEATURES
  - FAIL TO RECOGNIZE FEATURES OF Ada WILL INCREASE PROGRAMMER PRODUCTIVITY

INSTRUCTOR NOTES

- <u>FIRST STATEMENT</u>
  - SOFTWARE PEOPLE OFTEN MAKE GRAND PROMISES TO MANAGEMENT, AND THEN FAIL TO DELIVER. SOME MANAGERS MAY FEEL Ada IS JUST ANOTHER PROMISE THAT WILL BE BROKEN.
  - SOME MANAGERS MAY WANT TO LET OTHER PROJECTS LEARN HOW TO USE Ada, THEREBY BECOMING AN Ada SOURCE FOR HIS/HER PEOPLE. THIS MINIMIZES THE MANAGER'S RISK.

- <u>SECOND STATEMENT</u>
  - FIRST RELEASES OF COMPILERS HAVE TRADITIONALLY CONTAINED AN ENORMOUS NUMBER OF PROBLEMS PROGRAMMERS FREQUENTLY NEED TO SPEND TIME TO FIGURE OUT WAYS TO WORK AROUND COMPILER PROBLEMS.
  - MANAGERS NEED TO UNDERSTAND THE VALIDATION PROCESS AND THE ACVC.

PROBABLE RESISTANCE TO Ada: MANAGER'S VIEWPOINT - CONTINUED

"LET SOME OTHER PROJECT BE THE GUINA PIG"

- REASONS:
  - VICTIM OF SOFTWARE PROMISES THAT WEREN'T KEPT
  - CONTENT TO LET OTHER'S FIGURE OUT HOW TO USE Ada

"LET SOME OTHER PROJECT SHAKE OUT COMPILE BUGS."

- REASONS:
  - TRADITIONALLY, FIRST COMPILER RELEASES FULL OF BUGS
  - MATURE COMPILER TAKES A WHILE TO EVOLVE
  - DON'T UNDERSTAND ACVC ROLE IN VALIDATION

INSTRUCTOR NOTES

- THIS IS THE FIRST OF THREE SLIDES DESCRIBING SOME PREDICTABLE DIFFICULTIES IN TEACHING THE Ada CURRICULUM. EACH MODULE CLOSEUP WILL DESCRIBE PROBLEM AREAS PECULIAR TO THAT MODULE. THESE THREE DIFFICULTIES ARE JUST REPRESENTATIVE.

- REMEMBER THAT C, FORTRAN AND CMS HAVE NO NOTION OF STRONG TYPING. THUS THIS WILL BE AN ALIEN CONCEPT TO MANY PROGRAMMERS.

PREDICTABLE DIFFICULTIES: STRONG TYPING

- STRONG TYPING WILL BE DIFFICULT FOR PROGRAMMERS NOT USED TO DEALING WITH TYPING.
  - NO INTEGER TYPE TO FLOATING POINT TYPE CONVERSION
  - NO INTEGER TYPE TO INTEGER TYPE CONVERSION
  - CAN'T ADD INTEGER TO A POINTER
- SOME PROGRAMMERS WILL VIEW AS IMPEDIMENT TO EFFECTIVE PROGRAMMING

INSTRUCTOR NOTES

- LATER Ada MODULES ADDRESS THESE CONCERNS TO VARYING DEGREES.
- REAL TIME PROGRAMMERS WILL SEE A WIDE RANGE OF PROBLEMS AND THEIR SOLUTIONS IN L401, WHILE MANAGERS WILL SEE A SMALLER RANGE IN L303.
- IN PARTICULAR, L303 AND L401
  - COVER CYCLIC EXECUTIVES
  - DISCUSS PERFORMANCE THROUGH TUNING
    - L303 GIVES OVERVIEW
    - L401 IN-DEPTH COVERAGE
- L401 ALSO DISCUSSES WAYS TO REDUCE THE NUMBER OF TASKS.

PREDICTABLE DIFFICULTIES: MULTI-TASKING IN REAL TIME SYSTEMS

- LANGUAGE SUPPORT FOR MULTI-TASKING WILL BE NEW TO MOST PROGRAMMERS
- RENDEZVOUS AND SELECT STATEMENTS MAY NOT APPEAR FLEXIBLE ENOUGH TO HANDLE MULTI-TASKING NEEDS
- CONCERN FOR
  - CYCLIC EXECUTIVES
  - PERFORMANCE
  - NUMBER OF TASKS

INSTRUCTOR NOTES

- THIS DIFFICULTY MAY SHOW UP IN THE Ada MODULES AND IN THE METHODOLOGY MODULES.
- SOME PROGRAMMERS THINK THEY UNDERSTAND MODULARITY, BUT DO NOT.
- IN THE Ada COURSES, THIS WILL SHOW UP IN L305 EXERCISES.

PREDICTABLE DIFFICULTIES: MODULARITY

- MOST LANGUAGES HAVE VERY LITTLE SUPPORT FOR MODULARITY
- CONCEPTS SUCH AS:
  - HIGH COHESION
  - LOOSELY COUPLED
  - INFORMATION HIDING
  - DATA ABSTRACTION

  WILL BE DIFFICULT TO FULLY APPRECIATE

INSTRUCTOR NOTES

- THIS IS THE FIRST OF THREE SLIDES DESCRIBING SOME TYPICAL TRAPS TO AVOID WHEN TEACHING Ada FEATURES.
- OVERSELLING IS EASY TO FALL INTO BUT SHOULD BE AVOIDED.
- EVEN FOR A NON-SKEPTICAL AUDIENCE, OVERSELLING CAN MAKE THE INSTRUCTOR'S OBJECTIVITY SUSPECT.
- IT'S O.K. TO BE ENTHUSIASTIC!

TRAPS TO AVOID: OVERSELLING

- SOME AUDIENCES WILL BE HIGHLY SKEPTICAL
  - THOSE WHO HAVE NEVER USED HIGH LEVEL LANGUAGES
    - AVOID PROPAGANDA ABOUT HOW GREAT HIGH LEVEL LANGUAGES ARE OR YOU MIGHT ALIENATE THEM
  - REAL TIME PROGRAMMERS
    - AVOID TELLING THEM THAT EVERY REAL TIME PROBLEM CAN BE EASILY SOLVED IN Ada, OR YOU MIGHT BE ASKED TO SOLVE ONE
- FOR HIGHLY SKEPTICAL AUDIENCE
  - IF CAN'T WIN OVER SKEPTICS, THEN SETTLE FOR NEUTRALITY
  - IF COURSE DOES NOT CONVINCE THEM, EXPERIENCE PROBABLY WILL

INSTRUCTOR NOTES

- THIS SLIDE EXPLAINS WHY DESCRIBING AN Ada FEATURE IN TERMS OF ITS IMPLEMENTATION SHOULD BE AVOIDED.
  - MAY BE MISLEADING
    - EXPLAINING GENERIC INSTANTIATION AS MACRO EXPANSION IS INCORRECT
  - MAY CAUSE PROGRAMMERS TO AVOID FEATURE BECAUSE THEY FEEL IT IS NOT EFFICIENT.

- OTHER AREAS WHERE EXPLAINING Ada FEATURES COULD BE MISLEADING INCLUDE
  - GARBAGE COLLECTION
    - MAY BE DYNAMIC
    - MAY BE DETERMINED AT COMPILATION-TIME
    - MAY NOT BE PERFORMED AT ALL
  - TASKING
    - MANY OPTIMIZATIONS POSSIBLE (SOME DESCRIBED IN L401)
    - MAY REMOVE TASKING OVERHEAD COMPLETELY

TRAPS TO AVOID: DESCRIBING Ada FEATURES IN TERMS OF IMPLEMENTATION

- TEMPTING TO EXPLAIN Ada CONCEPTS IN TERMS OF "TYPICAL IMPLEMENTATION"
- EXAMPLE: GENERIC UNITS
  - MANY TIMES GENERIC INSTANTIATION EXPLAINED AS SIMILAR TO "MACRO" EXPANSION
  - SEEMS TO BE GOOD CONCEPTUAL CRUTCH
  - TOO MANY PEOPLE TAKE THIS LITERALLY
    - COMPLAIN THAT GENERIC PROGRAMMING WASTES SPACE
  - TRUTH IS, DEPENDS ON IMPLEMENTATION
    - SOME IMPLEMENTATIONS MIGHT ONLY USE "MACRO" APPROACH
    - SOME IMPLEMENTATIONS MIGHT ONLY USE RUNTIME DESCRIPTORS
    - SOME IMPLEMENTATIONS MIGHT BE HYBRID
  - REASONABLE IMPLEMENTATION OF GENERIC STACK MIGHT USE
    - SINGLE "EXPANSION" TO HANDLE STACK ITEMS THAT FIT IN ONE WORD
    - SINGLE "EXPANSION" WITH RUNTIME DESCRIPTORS FOR ALL OTHER CASES
    - VARY THIS IF SPACE/TIME OPTIMIZATION REQUESTED

INSTRUCTOR NOTES

- AVOID DISCUSSING EFFICIENCY UNTIL L401
- WHEN FORCED TO DISCUSS EFFICIENCY, STICK TO WHAT BULLET 4 SUGGESTS

TRAPS TO AVOID: EFFICIENCY

- EFFICIENCY IS AN IMPORTANT CONSIDERATION IN PROGRAMMING:
  - BUT NOT THE MOST IMPORTANT
  - JUST ONE OF MANY IMPORTANT CONSIDERATIONS
- PROGRAMS MUST BE
  - CORRECT
  - RELIABLE
  - MAINTAINABLE
- EFFICIENCY DISCUSSED FOR ENTIRE DAY IN L401
  - RESIST TALKING ABOUT IT BEFORE THEN
- IF FORCED TO DISCUSS IT SOONER REMEMBER
  - EXPERIENCE SHOWS ONLY SMALL PARTS OF A PROGRAM EFFECT EFFICIENCY
  - ALGORITHMS AND DATA STRUCTURES CAN HAVE MAJOR IMPACT ON EFFICIENCY
- DO NOT TELL STUDENTS THAT EFFICIENCY IS NOT A CONCERN

INSTRUCTOR NOTES

- EXERCISES ARE DISCUSSED IN MORE DEPTH IN EACH MODULE. THIS SLIDE JUST GIVES AN OVERVIEW OF HOW EXERCISES ARE ORGANIZED.

- BULLET 3 - ITEM 2

  - ONE OF THE AUTHOR'S OF THIS COURSE ENCOUNTERED AN L305 STUDENT WHO RUSHED TO DO ALL OF THE EXERCISES IN THE WORKBOOK. HOWEVER, THE STUDENT FAILED TO SHOW RECOGNITION OF THE MAIN POINTS OF THE EXERCISES. FOR EXAMPLE, THE STUDENT INCLUDED SUBPROGRAM SPECIFICATIONS AND TYPE AND OBJECT DECLARATIONS IN A PACKAGE SPECIFICATION, EVEN THESE WERE ONLY USED IN THE PACKAGE BODY.

EXERCISES IN THE Ada CURRICULUM

- IN-CLASS EXERCISES
  - IN THE LANGUAGE AND ENVIRONMENT MODULES THESE EXERCISES ARE USED TO MEASURE STUDENT UNDERSTANDING OF MATERIAL
  - IN METHODOLOGY MODULES ALL EXERCISES ARE IN-CLASS
    - MEASURE STUDENT PROGRESS
    - ALSO PROVIDE PRACTICAL EXPERIENCE
- HANDS-ON EXERCISES
  - EXERCISES GENERALLY PRACTICAL/EVERYDAY TYPE PROBLEMS
  - STUDENTS REQUIRED TO
    - SPECIFY INTERFACES
    - CONSIDER MODULARITY
    - USE Ada ENVIRONMENT
- EXERCISE WORKBOOKS INCLUDE MORE EXERCISES THAN NEEDED FOR A MODULE
  - PROVIDE INSTRUCTOR'S WITH CHOICES
  - WATCH OUT FOR THE STUDENT WHO
    - RUSHES TO COMPLETE EVERY EXERCISE IN A WORKBOOK
    - BUT MISSES THE POINT OF THE EXERCISES